



BEA WebLogic Java Adapter for Mainframe

Configuration and Administration Guide

BEA WebLogic Java Adapter for Mainframe 4.2
Document Edition 4.2
July 2001

Copyright

Copyright © 2001 BEA Systems, Inc. All Rights Reserved.

Restricted Rights Legend

This software and documentation is subject to and made available only pursuant to the terms of the BEA Systems License Agreement and may be used or copied only in accordance with the terms of that agreement. It is against the law to copy the software except as specifically allowed in the agreement. This document may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from BEA Systems, Inc.

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the BEA Systems License Agreement and in subparagraph (c)(1) of the Commercial Computer Software-Restricted Rights Clause at FAR 52.227-19; subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, subparagraph (d) of the Commercial Computer Software--Licensing clause at NASA FAR supplement 16-52.227-86; or their equivalent.

Information in this document is subject to change without notice and does not represent a commitment on the part of BEA Systems. THE SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FURTHER, BEA Systems DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE, OR THE RESULTS OF THE USE, OF THE SOFTWARE OR WRITTEN MATERIAL IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE.

Trademarks or Service Marks

BEA, WebLogic, Tuxedo, and Jolt are registered trademarks of BEA Systems, Inc. How Business Becomes E-Business, Operating System for the Internet, Liquid Data, BEA WebLogic E-Business Platform, BEA Builder, BEA Manager, BEA eLink, BEA Campaign Manager for WebLogic, BEA WebLogic Commerce Server, BEA WebLogic Personalization Server, BEA WebLogic Process Integrator, BEA WebLogic Collaborate, BEA WebLogic Enterprise, BEA WebLogic Server, and BEA WebLogic Integration are trademarks of BEA Systems, Inc.

All other trademarks are the property of their respective company.

BEA WebLogic Java Adapter for Mainframe Configuration and Administration Guide

Document Edition	Part Number	Date	Software Version
4.2	N/A	July 2001	BEA WebLogic Java Adapter for Mainframe 4.2

Contents

About This Document

What You Need to Know	viii
e-docs Web Site	viii
How to Print the Document	ix
Related Information	ix
Documentation Conventions	x

1. Preparing Mainframe Configurations for CRM Requirements

Action List	1-1
Prerequisites	1-2
How the CRM Works	1-3
Configuring the CRM for Windows NT or UNIX Platforms	1-4
Third-Party Stack Configuration	1-4
VTAM Cross Platform Definitions	1-12
XCA Major Node Defines the LAN Adapter for SYS1	1-12
Switched Network (SWNET) Definitions	1-13
Configuring the CRM for OS/390 Platforms	1-14
Configuring the CICS/ESA LU	1-15
Creating Connections at the Remote Host	1-15
Defining the Session at the Remote Host	1-15
Installing Resource Definitions	1-16
Viewing Connection and Session Status	1-16
What Do I Do Next?	1-17

2. Defining the JAM Gateway Configuration

Action List	2-1
Prerequisites	2-2

About the JAM Gateway Configuration File	2-2
Example of a JAM Gateway Configuration File	2-3
Format of the JAM Gateway Configuration File	2-4
JC_REMOTE_DOMAINS Section.....	2-5
JC_SNACRM Section.....	2-6
JC_SNASTACKS Section	2-8
JC_SNALINKS Section.....	2-9
JC_LOCAL_SERVICES Section	2-13
JC_REMOTE_SERVICES Section	2-15
Creating a Configuration File	2-19
Verifying the Configuration File with the Configuration Checker Utility	2-19
What Do I Do Next?	2-20

3. Putting It All Together

Action List	3-1
Prerequisites.....	3-2
Example of Windows NT or UNIX Cross-Platform Definitions	3-2
Example of OS/390 Definitions	3-5
What Do I Do Next?	3-8

4. Using the CRM Administration Commands

Action List	4-1
Prerequisites.....	4-2
Starting and Stopping the CRM.....	4-2
CRM Command.....	4-3
Command Line Options	4-3
Environment Variables.....	4-6
Diagnostics	4-6
Examples	4-6
Sample CRM Command JCL for OS/390 MVS Platform	4-7
crmdown Command	4-8
Command Line Options	4-9
Diagnostics.....	4-9
Example.....	4-9
Sample crmdown Command JCL for OS/390 MVS Platform.....	4-10

Starting the CRM on UNIX, OS/390 UNIX, or Windows NT	4-11
Starting the CRM on OS/390 MVS.....	4-12
Stopping the CRM On All Platforms	4-13
What Do I Do Next?.....	4-14

5. Deploying Your Configuration

Action List.....	5-1
Prerequisites	5-2
Placing the Gateway Configuration File in the WebLogic Server Start Directory.....	5-3
Setting the WebLogic Server CLASSPATH for the JAM.JAR File	5-3
Adding the JAM Startup Class to the WebLogic Server Configuration File	5-4
Verifying that the JAM Gateway is Ready.....	5-6
What Do I Do Next?.....	5-7

6. Using JAM Administration Utilities

Action List.....	6-1
Prerequisites	6-2
Activating and De-Activating Links	6-3
crlmkon Command.....	6-4
Command Line Options	6-4
Example	6-5
Diagnostics.....	6-5
Sample crlmkon Command JCL for OS/390 MVS Platform	6-5
crlmkoff Command	6-7
Command Line Options	6-8
Example	6-8
Diagnostics.....	6-8
Sample crlmkoff Command JCL for OS/390 MVS Platform.....	6-9
Using the CRM Monitor.....	6-10
Launching the CRM Monitor from the Windows NT Desktop	6-11
Launching the CRM Monitor from the Command Line	6-11
Setting CRM Monitor Options.....	6-11
Starting and Stopping the Gateway with the JAM Administration Servlet....	6-14
Launching the JAM Administration Servlet	6-15

Setting JAM Administration Servlet Options.....	6-16
Using JAM Administration Servlet Start and Stop Commands	6-16
What Do I Do Next?	6-17

7. Shutting Down the System for Planned Outages

Action List	7-1
Prerequisites.....	7-2
Shutting Down the System	7-3
Restarting the System	7-3
What Do I Do Next?	7-4

Index

About This Document

The BEA WebLogic Java Adapter for Mainframe product (hereafter referred to as JAM) is a gateway connectivity application that enables client/server interactions between Java applications and OS/390 Customer Information Control System/Enterprise System Architecture (CICS/ESA) or Information Management System (IMS) programs.

To use the BEA WebLogic Java Adapter for Mainframe (JAM) product with your system, you must set up and configure the following two major components:

- Communications Resource Manager (CRM) - prepares and receives message transmissions over connections to and from the mainframe
- JAM Gateway (JCRMGW) - processes Java-to-mainframe requests and responses, in conjunction with the CRM

After you configure the JAM product, you can use the administration utilities and tools to issue commands to the CRM, monitor the CRM, and issue administrative commands to the JAM gateway.

This document provides the following topics on JAM configuration and administration tasks:

- [“Preparing Mainframe Configurations for CRM Requirements”](#) describes mainframe configuration requirements for the CRM.
- [“Defining the JAM Gateway Configuration”](#) describes how to create the JAM configuration file and verify it.
- [“Putting It All Together”](#) shows an example of a JAM configuration on a mainframe operating system.
- [“Using the CRM Administration Commands”](#) describes how to start and stop the CRM.
- [“Deploying Your Configuration”](#) describes how to deploy the JAM configuration file and start your JAM system.
- [“Using JAM Administration Utilities”](#) describes how to use the CRM Monitor to set and display trace status and observe link status, and use the JAM

Administration Servlet to start or stop the gateway independently of WebLogic Server.

- “[Shutting Down the System for Planned Outages](#)” describes the process for shutting down parts or all of your JAM system when you have outages in your remote environment.
- “[Index](#)”

What You Need to Know

This document is intended for system administrators who will administer the BEA WebLogic Java Adapter for Mainframe application. Administrators should be familiar with basic administrative tasks for the system on which the JAM gateway is to run. They should also be familiar with, or have access to administrators who are familiar with mainframe administration, including Virtual Telecommunications Access Method (VTAM).

e-docs Web Site

BEA product documentation is available on the BEA corporate Web site. From the BEA Home page, click on Product Documentation or go directly to the “e-docs” Product Documentation page at <http://edocs.bea.com/>.

How to Print the Document

A PDF version of this document is available on the JAM documentation Home page on the e-docs Web site and also on the documentation CD included with your installation CD. You can open the PDF in Adobe Acrobat Reader and print the entire

document (or a portion of it) in book format. To access the PDFs, open the JAM documentation Home page, click the PDF files button, and select the document you want to print.

If you do not have the Adobe Acrobat Reader, you can get it for free from the Adobe Web site at <http://www.adobe.com/>.

Related Information

The following BEA publications are available for JAM 4.2:

- *BEA WebLogic Java Adapter for Mainframe Release Notes*
- *BEA WebLogic Java Adapter for Mainframe Introduction*
- *BEA WebLogic Java Adapter for Mainframe Installation Guide*
- *BEA WebLogic Java Adapter for Mainframe Configuration and Administration Guide*
- *BEA WebLogic Java Adapter for Mainframe Programming Guide*
- *BEA WebLogic Java Adapter for Mainframe Scenarios Guide*
- *BEA WebLogic Java Adapter for Mainframe Workflow Processing Guide*
- *BEA WebLogic Java Adapter for Mainframe Reference Guide*

Contact Us

Your feedback on the BEA WebLogic Java Adapter for Mainframe documentation is important to us. Send us e-mail at docsupport@bea.com if you have questions or comments. Your comments will be reviewed directly by the BEA professionals who create and update the JAM documentation.

In your e-mail message, please indicate that you are using the documentation for the BEA WebLogic Java Adapter for Mainframe 4.2 release.

If you have any questions about this version of JAM, or if you have problems installing and running JAM, contact BEA Customer Support through BEA WebSupport at **www.bea.com**. You can also contact Customer Support by using the contact information provided on the Customer Support Card that is included in the product package.

When contacting Customer Support, be prepared to provide the following information:

- Your name, e-mail address, phone number, and fax number
- Your company name and company address
- Your machine type and authorization codes
- The name and version of the product you are using
- A description of the problem and the content of pertinent error messages

Documentation Conventions

The following documentation conventions are used throughout this document.

Convention	Item
blue text	Indicates a hypertext link in PDF or HTML
<i>italics</i>	Indicates emphasis or book titles or variables.
"string with quotes"	Indicates a string entry that requires quote marks.

Convention	Item
UPPERCASE TEXT	Indicates generic file names, device names, environment variables, and logical operators. <i>Examples:</i> LPT1 SIGNON OR
monospace text	Indicates code samples, commands and their options, data structures and their members, data types, directories, and file names and their extensions. Monospace text also indicates text that you must enter from the keyboard. <i>Examples:</i> #include <iostream.h> void main () the pointer psz chmod u+w * \tux\data\ap .doc tux.doc BITMAP float
monospace boldface text	Identifies significant words in code. <i>Example:</i> void xa_commit ()
{ }	Indicates a set of choices in a syntax line. The braces themselves should never be typed.
[]	Indicates optional items in a syntax line. The brackets themselves should never be typed. <i>Example:</i> buildclient [-v] [-o name] [-f file-list]... [-l file-list]...
	Separates mutually exclusive choices in a syntax line. The symbol itself should never be typed.

Convention	Item
...	<p>Indicates one of the following in a command line:</p> <ul style="list-style-type: none"> ■ That an argument can be repeated several times in a command line ■ That the statement omits additional optional arguments ■ That you can enter additional parameters, values, or other information <p>The ellipsis itself should never be typed.</p> <p><i>Example:</i></p> <pre>buildclient [-v] [-o name] [-f file-list]... [-l file-list]...</pre>
.	<p>Indicates the omission of items from a code example or from a syntax line.</p> <p>The vertical ellipsis itself should never be typed.</p>

1 Preparing Mainframe Configurations for CRM Requirements

The Communications Resource Manager (CRM) runs as a separate native process that provides an emulation allowing Customer Information Control System/Enterprise System Architecture (CICS/ESA) and Information Management System (IMS) protocols to flow into and out of the Java environment. After you install the WebLogic Java Adapter for Mainframe (JAM) product, you must set up your mainframe configurations to allow the CRM to access your CICS or IMS systems.

If the CRM runs on the mainframe, it uses the native SNA networking product, Virtual Telecommunications Access Method (VTAM), to establish its SNA connectivity to the CICS or IMS application environments. If the CRM runs on a Windows NT or UNIX platform, a third-party SNA stack must also be installed on this platform and configured to establish SNA connections with the CRM and target mainframe.

The topics in this section cover mainframe configuration requirements for the CRM.

Action List

Before you prepare mainframe configurations for CRM requirements, see the following action list and refer to the appropriate information sources.

1 Preparing Mainframe Configurations for CRM Requirements

Your action...	Refer to...
1 Complete all prerequisite tasks.	“Prerequisites”
2 Learn about the CRM.	“How the CRM Works”
3 Learn about configurations for Windows NT or UNIX platforms.	“Configuring the CRM for Windows NT or UNIX Platforms”
4 Learn about configurations for OS/390 platforms.	“Configuring the CRM for OS/390 Platforms”
5 Learn about configurations for the CICS/ESA LU..	“Configuring the CICS/ESA LU”
6 Prepare for the next step.	“What Do I Do Next?” and the next section, “Defining the JAM Gateway Configuration”

Prerequisites

Before you prepare mainframe configurations for CRM requirements, you should complete the following tasks:

Your action...	Refer to...
1 Verify that system requirements have been met.	<i>BEA WebLogic Java Adapter for Mainframe Introduction and Release Notes</i>
2 Determine which configuration your system requires and install the JAM product as appropriate for your configuration.	<i>BEA WebLogic Java Adapter for Mainframe Installation Guide</i>

How the CRM Works

The CRM is the component of the BEA WebLogic Java Adapter for Mainframe (JAM) that manages communications resources. The CRM coordinates the flow of data between Java applications running on a WebLogic Server platform and applications running on a mainframe. The mainframe applications may use CICS/ESA with Distributed Program Link (DPL) or IMS with implicit Advanced Program-to-Program Communications (APPC).

The CRM uses both SNA and TCP communication protocols. TCP protocols always flow between the JAM gateway and CRM. SNA protocols always flow from the CRM to the mainframe and from either an SNA stack or VTAM on the mainframe to the CRM depending upon the configuration option.

The CRM may be located on the same machine as the JAM gateway or distributed on another UNIX, Windows NT, or mainframe operating system. For a complete list of operating systems, refer to the *BEA WebLogic Java Adapter for Mainframe Release Notes*. If the CRM is not running on a mainframe, it must run on the same platform as the SNA stack.

The following requirements should be considered for establishing a VTAM Configuration:

- If you are running your CRM on a mainframe, your mainframe must have a local VTAM configuration that communicates with the CRM.
- If your JAM system is used in a VTAM environment, make sure the host configuration supports VTAM.
- If you are using a third-party SNA stack, you must configure VTAM to communicate with the SNA stack.

Consult with your CICS/ESA remote domain administrator to obtain key parameters in the VTAM definition that must be included in the SNA stack configuration, as well as in other configuration files in the JAM local domain.

Configuring the CRM for Windows NT or UNIX Platforms

A basic understanding of the mainframe configuration requirements provides a context for understanding the CRM functions and configuration requirements. The following sections discuss the various configuration considerations and give examples of those configurations.

Note: Consult with your local mainframe system administrator for specific information about your system. The examples in the following sections illustrate a starting point for configuring your system and do not represent all possibilities. The examples represent one way a mainframe can be configured to work in an Advanced Peer-to-Peer Networking (APPN) Local Area Network (LAN) environment.

Third-Party Stack Configuration

Third-party stack configurations may be set up using stack-specific configuration utilities or by manually creating a configuration file with any text editor. [Listing 1-1](#) shows an example of an SNA node configuration file. The configuration file is divided into sections for various components of the configuration. Each section defines a component using parameters (or keywords) and values. Some of these keywords and values will affect how your JAM configuration is defined.

Listing 1-1 SNA Node Configuration File

```
[define_node_config_file]
major_version = 5
minor_version = 1
update_release = 1
revision_level = 57

[define_node]
cp_alias = dqasun1
description = ""
```



```
fqcp_name = BEALAN.DQASUN1
node_type = LEN_NODE
mode_to_cos_map_supp = YES
mds_supported = YES
node_id = <bea12899>
max_locates = 1500
dir_cache_size = 255
max_dir_entries = 0
locate_timeout = 0
reg_with_nn = YES
reg_with_cds = YES
mds_send_alert_q_size = 100
cos_cache_size = 24
tree_cache_size = 40
tree_cache_use_limit = 40
max_tdm_nodes = 0
max_tdm_tgs = 0
max_isr_sessions = 1000
isr_sessions_upper_threshold = 900
isr_sessions_lower_threshold = 800
isr_max_ru_size = 16384
isr_rcv_pac_window = 8
store_endpt_rscvs = NO
store_isr_rscvs = NO
store_dlur_rscvs = NO
cos_table_version = VERSION_0_COS_TABLES
send_term_self = NO
disable_branch_awareness = NO
cplu_syncpt_support = NO
cplu_attributes = NONE
dlur_support = NO
pu_conc_support = YES
nn_rar = 128
max_ls_exception_events = 0
ptf_flags = NONE

[define_ethernet_dlc]
dlc_name = ETHER0
description = ""
neg_ls_supp = YES
initially_active = NO
adapter_number = 0
lan_type = 802_3_DIX

[define_ethernet_port]
port_name = ETSAP0
description = ""
dlc_name = ETHER0
port_type = PORT_SATF
```

1 *Preparing Mainframe Configurations for CRM Requirements*

```
port_number = 0
max_rcv_btu_size = 1033
tot_link_act_lim = 64
inb_link_act_lim = 0
out_link_act_lim = 0
ls_role = LS_NEG
implicit_dspu_services = NONE
implicit_dspu_template = ""
implicit_ls_limit = 0
act_xid_exchange_limit = 9
nonact_xid_exchange_limit = 5
ls_xmit_rcv_cap = LS_TWS
max_ifrm_rcvd = 7
target_pacing_count = 7
max_send_btu_size = 1033
mac_address = <080020a6808a>
lsap_address = 0x04
implicit_cp_cp_sess_support = NO
implicit_limited_resource = NO
implicit_deact_timer = 30
implicit_hpr_support = NO
implicit_link_lvl_error = NO
implicit_uplink_to_en = NO
effect_cap = 3993600
connect_cost = 0
byte_cost = 0
security = SEC_NONSECURE
prop_delay = PROP_DELAY_LAN
user_def_parm_1 = 128
user_def_parm_2 = 128
user_def_parm_3 = 128
initially_active = YES
window_inc_threshold = 1
test_timeout = 10
test_timer_retry = 5
xid_timer = 10
xid_timer_retry = 5
ack_timeout = 5000
p_bit_timeout = 5000
t2_timeout = 100
rej_timeout = 10
busy_state_timeout = 30
idle_timeout = 30
max_retry = 3

[define_ethernet_ls]
ls_name = DV10QSN1
description = ""
port_name = ETSAP0
```

```
adj_cp_name = P390.DALVS10
adj_cp_type = END_NODE
mac_address = <0200bea07002>
lsap_address = 0x08
auto_act_supp = NO
tg_number = 0
limited_resource = NO
solicit_sscp_sessions = YES
pu_name = DV10QSN1
disable_remote_act = NO
default_nn_server = NO
dspu_services = NONE
dspu_name = <000000000000000000>
dlus_name = <0000000000000000000000000000000000000000>
bkup_dlus_name = <0000000000000000000000000000000000000000>
hpr_supported = NO
hpr_link_lvl_error = NO
link_deact_timer = 30
use_default_tg_chars = YES
ls_attributes = SNA
adj_node_id = <00000000>
local_node_id = <00000000>
cp_cp_sess_support = NO
effect_cap = 3993600
connect_cost = 0
byte_cost = 0
security = SEC_NONSECURE
prop_delay = PROP_DELAY_LAN
user_def_parm_1 = 128
user_def_parm_2 = 128
user_def_parm_3 = 128
target_pacing_count = 7
max_send_btu_size = 1033
ls_role = USE_PORT_DEFAULTS
max_ifrm_rcvd = 0
dlus_retry_timeout = 0
dlus_retry_limit = 0
branch_link_type = NONE
adj_brnn_cp_support = ALLOWED
dddlu_offline_supported = NO
initially_active = NO
restart_on_normal_deact = NO
react_timer = 30
react_timer_retry = 65535
test_timeout = 10
test_timer_retry = 5
xid_timer = 10
xid_timer_retry = 5
ack_timeout = 5000
```

1 *Preparing Mainframe Configurations for CRM Requirements*

```
p_bit_timeout = 5000
t2_timeout = 100
rej_timeout = 10
busy_state_timeout = 30
idle_timeout = 30
max_retry = 3

[define_ethernet_ls]
ls_name = DVS5QSN1
description = ""
port_name = ETSAP0
adj_cp_name = P390.DALVS5
adj_cp_type = END_NODE
mac_address = <0200bea05004>
lsap_address = 0x08
auto_act_supp = NO
tg_number = 0
limited_resource = NO
solicit_sscp_sessions = YES
pu_name = DVS5QSN1
disable_remote_act = NO
default_nn_server = NO
dspu_services = NONE
dspu_name = <0000000000000000>
dlus_name = <0000000000000000000000000000000000>
bkup_dlus_name = <0000000000000000000000000000000000>
hpr_supported = NO
hpr_link_lvl_error = NO
link_deact_timer = 30
use_default_tg_chars = YES
ls_attributes = SNA
adj_node_id = <00000000>
local_node_id = <bea12899>
cp_cp_sess_support = NO
effect_cap = 3993600
connect_cost = 0
byte_cost = 0
security = SEC_NONSECURE
prop_delay = PROP_DELAY_LAN
user_def_parm_1 = 128
user_def_parm_2 = 128
user_def_parm_3 = 128
target_pacing_count = 7
max_send_btu_size = 1033
ls_role = USE_PORT_DEFAULTS
max_ifrm_rcvd = 0
dlus_retry_timeout = 0
dlus_retry_limit = 0
branch_link_type = NONE
```

```
adj_brnn_cp_support = ALLOWED
dddlu_offline_supported = NO
initially_active = NO
restart_on_normal_deact = NO
react_timer = 30
react_timer_retry = 65535
test_timeout = 10
test_timer_retry = 5
xid_timer = 10
xid_timer_retry = 5
ack_timeout = 5000
p_bit_timeout = 5000
t2_timeout = 100
rej_timeout = 10
busy_state_timeout = 30
idle_timeout = 30
max_retry = 3
```

[define_partner_lu]

```
plu_alias = C10QA2
description = ""
fqplu_name = P390.C10QA2
plu_un_name = C10QA2
parallel_sess_supp = YES
max_mc_ll_send_size = 0
conv_security_ver = NO
```

[define_local_lu]

```
lu_alias = LUQASN1A
list_name = ""
description = ""
lu_name = LUQASN1A
lu_session_limit = 0
pu_name = <0000000000000000>
nau_address = 0
default_pool = NO
syncpt_support = YES
lu_attributes = NONE
sscp_id = 0
disable = NO
sys_name = ""
timeout = 60
back_level = NO
```

[define_local_lu]

```
lu_alias = LUQASN1B
list_name = ""
description = ""
lu_name = LUQASN1B
```

1 *Preparing Mainframe Configurations for CRM Requirements*

```
lu_session_limit = 0
pu_name = <0000000000000000>
nau_address = 0
default_pool = NO
syncpt_support = YES
lu_attributes = NONE
sscp_id = 0
disable = NO
sys_name = ""
timeout = 60
back_level = NO
```

```
[define_local_lu]
lu_alias = LUQASN1C
list_name = ""
description = ""
lu_name = LUQASN1C
lu_session_limit = 0
pu_name = <0000000000000000>
nau_address = 0
default_pool = NO
syncpt_support = YES
lu_attributes = NONE
sscp_id = 0
disable = NO
sys_name = ""
timeout = 60
back_level = NO
```

```
[define_local_lu]
lu_alias = LUQASN1D
list_name = ""
description = ""
lu_name = LUQASN1D
lu_session_limit = 0
pu_name = <0000000000000000>
nau_address = 0
default_pool = NO
syncpt_support = YES
lu_attributes = NONE
sscp_id = 0
disable = NO
sys_name = ""
timeout = 60
back_level = NO
```

```
[define_mode]
mode_name = SMSNA100
description = ""
```

```
max_neg_sess_lim = 32767
plu_mode_session_limit = 10
min_conwin_src = 5
min_conloser_src = 5
auto_act = 5
receive_pacing_win = 4
max_receive_pacing_win = 0
default_ru_size = YES
max_ru_size_upp = 1024
max_ru_size_low = 0
cos_name = #CONNECT

[define_directory_entry]
resource_name = P390.DALVS5
resource_type = ENCP_RESOURCE
description = (Auto defined - remote node)
parent_name = <00000000000000000000000000000000>
parent_type = ENCP_RESOURCE

[define_directory_entry]
resource_name = P390.DALVS5
resource_type = LU_RESOURCE
description = (Auto defined - default LU)
parent_name = P390.DALVS5
parent_type = ENCP_RESOURCE

[define_directory_entry]
resource_name = P390.DALVS10
resource_type = ENCP_RESOURCE
description = (Auto defined - remote node)
parent_name = <00000000000000000000000000000000>
parent_type = ENCP_RESOURCE

[define_directory_entry]
resource_name = P390.C10QA2
resource_type = LU_RESOURCE
description = ""
parent_name = P390.DALVS10
parent_type = ENCP_RESOURCE

[define_directory_entry]
resource_name = P390.DALVS10
resource_type = LU_RESOURCE
description = (Auto defined - default LU)
parent_name = P390.DALVS10
parent_type = ENCP_RESOURCE

[define_defaults]
description = ""
```

1 *Preparing Mainframe Configurations for CRM Requirements*

```
mode_name = SMSNA100
implicit_plu_forbidden = YES
specific_security_codes = NO
limited_timeout = 20
```

VTAM Cross Platform Definitions

The examples of cross platform definitions discussed in the following sections allow the VTAM network to communicate with your stack using the SNA protocol.

XCA Major Node Defines the LAN Adapter for SYS1

This definition is set up for use with an emulated IBM 3172 Interconnect Controller for connecting an APPN network node to another APPN node. Note that the definition is for an Ethernet LAN, and the `SAPADDR` specified must be the same as the `LSap` specified for the local link station.

Listing 1-2 XCA Major Node

```
XETH2LP1 VBUILD TYPE=XCA ** EXTERNAL COMMUNICATION ADAPT**
PORTE2   PORT  ADAPNO=1,      ** 3172 RELATIVE ADAPTER NUMBER**
          CUADDR=E22,        ** CHANNEL UNIT ADDRESS      **
          MEDIUM=CSMACD,    ** LAN TYPE=ETHERNET        **
          SAPADDR=8,        ** SERVICE ACCESS POINT ADDRESS**
          TIMER=120         ** CHANNEL ACTIVATE RESP TIME **
*
G1ETH2   GROUP DIAL=YES,      ** YES required for putype 2  **
          DYNPU=YES,
          CALL=INOUT,
          ANSWER=ON,
          ISTATUS=ACTIVE

LETH20   LINE
PETH20   PU
LETHE3   LINE
PETHE3   PU
LETHF3   LINE
PETHF3   PU
```

Switched Network (SWNET) Definitions

VTAM Switched Major Node (SWNET) definitions define and link together physical units (PU) and logical units (LU). Each workstation connected to a network must be represented on the network as a PU and each PU can have one or more LUs. The VTAM SWNET definition defines how the PUs and LUs communicate with a server.

The two switched network definition examples in this section ([Listing 1-3](#) and [Listing 1-4](#)) specify the VTAM PU, representing the local link stations that expect to connect with the host machine. The IDBLK and IDNUM definitions are provided to support 3270 traffic and must be unique, as well as match the values specified in the local link definition.

Listing 1-3 SWNET Major Node

```
SWNETHHP  VBUILD  TYPE=SWNET,MAXNO=3,MAXGRP=3
P390HP10  PU  ADDR=02,
          IDBLK=05F,
          IDNUM=FFFFF,
          PUTYPE=2,
          NETID=BEALAN,
          CPNAME=DALHP10,
          MAXPATH=3,
          DWACT=YES,
          CONNTYPE=APPN,
          CPCP=YES,
          DYNLU=YES
* -----
* SNA SAP & HP10 MAC ADDRESS BIT REVERSED FOR TRFMT
* -----
PATHHP    PATH  DIALNO=00041000900C24EE,
          GRPNM=G1ETH2
LUHP10A  LU  LOCADDR=0
LUHP10B  LU  LOCADDR=0
LUHP10C  LU  LOCADDR=0
```

Listing 1-4 SWNET Major Node

```
SWNETH2      VBUILD  TYPE=SWNET,MAXNO=3,MAXGRP=3
P390ETH2     PU   ADDR=04,
              IDBLK=019,
              IDNUM=10092,
              PUTYPE=2,
              NETID=BEALAN,
              CPNAME=SUN2,
              MAXPATH=3,
              DWACT=YES,
              CONNTYPE=APPN,
              CPCP=YES,
              DYNLU=YES

* -----
* SNA SAP & SUN2 MAC ADDRESS BIT REVERSED FOR TRFMT
* -----

PATH01      PATH  DIALNO=00081000043EE20A,
              GRPNM=G1ETH2

LUSUN2A LU   LOCADDR=0
LUSUN2B LU   LOCADDR=0
LUSUN2C LU   LOCADDR=0
```

Configuring the CRM for OS/390 Platforms

VTAM must be configured to allow the CRM to communicate with CICS or IMS using the SNA protocol.

The APPLID definition shown in [Listing 1-5](#) defines the local stack configuration to run under OS/390 using VTAM.

Listing 1-5 APPLID Definition (OS/390)

```
BEASNA VBUILD TYPE=APPL
BEAAPPL1 APPL ACBNAME=BEAAPPL1,
              AUTH=(ACQ,PASS),
              APPC=YES,
              SYNCLVL=CONFIRM,
              PARSESS=YES
```

Configuring the CICS/ESA LU

Before you can connect the CRM to the remote stack, the CICS/ESA LU (logical unit) configuration must be established. To establish the configuration, create connection definitions, create session definitions, and install resource definitions.

Creating Connections at the Remote Host

If a remote connection definition file is not already in place, work with the mainframe support personnel to create one. When placed on the remote host, the definition provides a connection with the local domain. Note the following example of a JAM connection definition file:

```
DEFINE CONNECTION (BEA)          GROUP (BEACONN)
    DE (JAM EXAMPLE RDO CONNECTION)
    ACCESSMETHOD (VTAM)          PROTOCOL (APPC)
    NETNAME (**VTAM NETWORK NAME OF REMOTE SYSTEM**)
    ATTACHSEC (LOCAL)           AUTOCONNECT (NO)
```

Defining the Session at the Remote Host

If a session definition is not already in place, work with the mainframe support personnel to create one. When placed on the remote host, the session definition defines the logical links by which the local domain communicates with the remote host. Note the following example of a JAM session definition:

```
DEFINE SESSION (BEATEST)         GROUP (BEACONN)
    CONNECTION (BEA)
    DE (JAM EXAMPLE RDO SESSION)
    PROTOCOL (APPC)              AUTOCONNECT (YES)
    MODENAME (**MODE**)          MAXIMUM (**SESSNBR**, **WINNER**)
```

The arguments and options in this example are defined in the following way:

1 *Preparing Mainframe Configurations for CRM Requirements*

AUTOCONNECT

Indicates how the activation of the session is negotiated.

YES

Enables the CICS/ESA host to negotiate its own winner sessions when a conversation is allocated.

MODENAME

Indicates either a CICS/ESA-supplied mode name, such as `SMSNA100`, or your own defined mode name. If another set of session definitions exists for the BEA connection, this mode name must be unique among all sets defined to the connection. The mode name corresponds to the VTAM `LOGMODE` name.

MAXIMUM

Defines the total number of sessions in the set and the total number of winner sessions. The total number of winner sessions must include those for the host and the remote stack. The `WINNER` number plus the number of remote sessions should equal the `SESSNR`.

Installing Resource Definitions

To install the resource definitions, put them on the host in a separate group. Use the `CEDA INSTALL` command.

For example:

```
CEDA INSTALL GROUP (BEACONN)
```

Viewing Connection and Session Status

After you have installed the resource definitions, you can view the status of connections and sessions using the following CICS/ESA system commands:

```
CEMT I CONN (BEA)           **view the status of the connection
CEMT I NET (**NETNAME**)    **View the status of the sessions
CEMT I MODENAME (**MODE**)  **View the status of the mode
```

What Do I Do Next?

After you prepare your mainframe configurations for the CRM, you are ready to create your JAM configuration file. Refer to [“Defining the JAM Gateway Configuration”](#) for more information about creating the configuration file.

1 *Preparing Mainframe Configurations for CRM Requirements*

2 Defining the JAM Gateway Configuration

BEA WebLogic Java Adapter for Mainframe (JAM) configuration is defined by the JAM configuration file, `jcrmgw.cfg`. The JAM gateway uses the `jcrmgw.cfg` file to control much of its operation. The `jcrmgw.cfg` configuration file defines the Communications Resource Manager (CRM), stack, links, and local and remote services that comprise the gateway environment.

The topics in this section describe the JAM gateway configuration file and how to create and verify your own configuration file.

Action List

Before you create your JAM gateway configuration file, see the following action list and refer to the appropriate information sources.

	Your action...	Refer to...
1	Complete all prerequisite tasks.	“Prerequisites”
2	Learn about the JAM gateway configuration file.	“About the JAM Gateway Configuration File”
3	Create a JAM gateway configuration file for your environment.	“Creating a Configuration File”

2 Defining the JAM Gateway Configuration

Your action...	Refer to...
4 Check the JAM gateway configuration file.	“Verifying the Configuration File with the Configuration Checker Utility”
5 Prepare for the next step.	“What Do I Do Next?” and the next section, “Putting It All Together”

Prerequisites

Before you create your JAM gateway configuration file, you should complete the following tasks:

Your action...	Refer to...
1 Verify that system requirements have been met.	<i>BEA WebLogic Java Adapter for Mainframe Introduction and Release Notes</i>
2 Determine which configuration your system requires and install the JAM product as appropriate for your configuration.	<i>BEA WebLogic Java Adapter for Mainframe Installation Guide</i>
3 Learn about mainframe configuration requirements for the CRM.	“Preparing Mainframe Configurations for CRM Requirements”

About the JAM Gateway Configuration File

A sample `jcrmgw.cfg` file is located in the JAM `examples/samples.jar` file. Edit this file to meet your configuration needs or create your own configuration file with any text editor. Refer to the following sections for information about each configuration file section and associated parameters.

Example of a JAM Gateway Configuration File

The following example illustrates a basic `jcrmgw.cfg` file.

Listing 2-1 Sample `jcrmgw.cfg` Configuration File

```
*JC_REMOTE_DOMAINS
#
CICS13      DOMAINID="13"
#
*JC_SNACRM
#
CRMAN      SNACRMADDR="//da1nt66:8650"
           GROUP="G1"

*JC_SNASTACKS
#
OS390      STACKTYPE="VTM28"
           LOCALLU="LUNT66B"
#
*JC_SNALINKS
#
CICS       RLUNAME="C410XB01"
           RDOM="CICS13"
           MODENAME="SMSNA100"
           MAXSESS=8
           MINWIN=6
#
*JC_LOCAL_SERVICES
#
TraderHome  RNAME="DPL1SVR"
#
*JC_REMOTE_SERVICES
DPLINIT     RDOM="CICS13"
           RNAME="PRIM:DPLINIT"
           TRANTIME=10000
TOUPPER     RDOM="CICS13"
           RNAME="TOUPDPLS"
           TRANTIME=10000
demoRead    RDOM="CICS13"
           RNAME="DPLDEMOR"
           TRANTIME=10000
demoUpdate  RDOM="CICS13"
           RNAME="DPLDEMOU"
           TRANTIME=10000
```

```
demoCreate      RDOM="CICS13"
                 RNAME="DPLDEMOC"
                 TRANTIME=10000
demoDelete      RDOM="CICS13"
                 RNAME="DPLEMOD"
                 TRANTIME=10000
imsInsert       RDOM="CICS13"
                 FUNCTION=APPC
                 RNAME="DPLDEMOC"
```

Format of the JAM Gateway Configuration File

The general format of the `jcrmgw.cfg` configuration file is as follows:

- The file is made up of six sections containing parameters. Lines beginning with an asterisk (*) indicate the beginning of a specific section. Each section title line contains the name of the section immediately following the asterisk. The following sections are defined:
 - [JC_REMOTE_DOMAINS Section](#)
 - [JC_SNACRM Section](#)
 - [JC_SNASTACKS Section](#)
 - [JC_SNALINKS Section](#)
 - [JC_LOCAL_SERVICES Section](#)
 - [JC_REMOTE_SERVICES Section](#)
- Parameters are generally specified by: *KEYWORD = value*. This sets *KEYWORD* to *value*. Valid keywords are described in the following sections. Keywords are reserved. They cannot be used as values unless they are quoted.
- Input fields are separated by at least one space (or tab) character.
- Each section may include sets of keywords for multiple domains, CRMs, stacks, links, home interfaces, or services. Therefore, each set of keywords must be identified with a label. The label is placed before the first keyword of the set of keywords as shown in the following example of two remote services labeled `DPLINIT` and `TOUPPER`:

```
*JC_REMOTE_SERVICES
DPLINIT   RDOM="CICS410"
          RNAME="PRIM:DPLINIT"
          TRANTIME=10000
TOUPPER   RDOM="CICS410"
          RNAME="TOUPDPLS"
          TRANTIME=10000
```

- “#” introduces a comment. A new line ends a comment.
- Blank lines and comments are ignored.
- Comments can be freely attached to the end of any line.

JC_REMOTE_DOMAINS Section

This section of the `jcrmgw.cfg` file provides an alias for associating mainframe applications with services and links. A label identifying the domain must precede the first keyword in the set of keywords defining the domain.

[Listing 2-2](#) shows an example of the `JC_REMOTE_DOMAINS` section. `CICS410` is the label for the domain.

Listing 2-2 Example of JC_REMOTE_DOMAINS Section

```
*JC_REMOTE_DOMAINS
#
CICS410      DOMAINID="410"
```

The following table provides descriptions of valid keywords for the `JC_REMOTE_DOMAINS` section:

2 Defining the JAM Gateway Configuration

Keyword	Default	Required/ Optional	Description
DOMAINID	None	Required	Name of a partner system DOMAINID=<string> <string> is any name to be used for identifying a partner system. Example: CRCICS1 DOMAINID="TestCICS"

JC_SNACRM Section

This section of the `jcrmgw.cfg` file identifies the CRM that this gateway talks to. There is one CRM for each JAM gateway. A label identifying the CRM must precede the first keyword.

[Listing 2-3](#) shows an example of the JC_SNACRM section. CRMAN is the label for the CRM.

Listing 2-3 Example of JC_SNACRM Section

```
*JC_SNACRM
#
CRMAN          SNACRMADDR="//da1nt66:8650"
                GROUP="G1"
```

The following table provides descriptions of valid keywords for the JC_SNACRM section:

Keyword	Default	Required/Optional	Description
SNACRMADDR	None	Required	<p>Symbolic TCP address</p> <p>SNACRMADDR=<string></p> <p><string> is a symbolic TCP address in the form of: <code>//hostname:port</code></p> <p>hostname is the name of the machine that runs the CRM. Port is an available decimal port number that the CRM uses to talk to the Java gateway. In the case of a CRM that is started independently of the gateway, this address must match the address used on the CRM command line. When the gateway is started, it tries to contact the CRM at the address.</p> <p>Example:</p> <pre>MYCRM SNACRMADDR="//dalhp55:6942"</pre> <p>The gateway will look for a CRM on a machine named dalhp55 at port 6942.</p>
GROUP	None	Required	<p>Name of group correlating gateway with CRM</p> <p>GROUP=<string></p> <p><string> is any name to be used to correlate a gateway with a CRM. The name is used as part of the file name for the CRM logs. In the case of a CRM that is started independently of the gateway, this name must match the group name used on the CRM command line, even if the default name is used. The keyword/value pair has a default value of GROUP.</p> <p>Example:</p> <pre>GROUP="CRAUTH"</pre> <p>The CRM expects a gateway signon for group CRAUTH.</p>

JC_SNASTACKS Section

This section of the `jcrmgw.cfg` file identifies the Local LU used for the CRM along with the stack identifier for the stack library to be used. Only one local LU and one stack can be specified for a CRM. A label identifying the Local LU and stack must precede the first keyword in the set of keywords defining the Local LU and stack.

[Listing 2-4](#) shows an example of the JC_SNASTACKS section. OS390ST is the label for the Local LU and stack.

Listing 2-4 Example of JC_SNASTACKS Section

```
*JC_SNASTACKS
#
OS390ST          STACKTYPE="VTM28"
                  LOCALLU="BEAAPPL1"
```

The following table provides descriptions of valid keywords for the JC_SNASTACKS section:

Keyword	Default	Required/ Optional	Description
LOCALLU	None	Required	Alias of local LU to be used with CRM LOCALLU=<string> <string> is an alias for the local LU to be used by the CRM. This must match a corresponding LU alias defined in the Stack configuration. This alias may or may not match the actual LU name. Example: LOCALLU="AUTHAPP" The CRM tries to use the local LU which has been defined in the stack configuration with an alias of AUTHAPP.

Keyword	Default	Required/ Optional	Description
STACKTYPE	None	Required	<p>Name of stack support library to load</p> <p>STACKTYPE= [ibm60 spx62 vt28]</p> <p>One of the specified tokens must be used. These names determine which stack support library will be loaded.</p> <p>The following stacks can be identified:</p> <ul style="list-style-type: none"> ■ <code>ibm60</code> = IBM Communications server 6.0 on WINNT ■ <code>spx62</code> = Data Connection Snapix 6.2 or 7.0 on Solaris ■ <code>vt28</code> = VTAM on OS/390 <p>Example:</p> <p><code>STACKTYPE="vt28"</code></p> <p>The CRM loads the library for VTAM on OS/390.</p>

JC_SNALINKS Section

This section of the `jcrmgw.cfg` file identifies partner mainframe application regions. Multiple links for a single CRM are supported. A label identifying the link must precede the first keyword in the set of keywords defining the link.

[Listing 2-5](#) shows an example of the `JC_SNALINKS` section. `C41XB01` is the label for the link.

Listing 2-5 Example of JC_SNALINKS Section

```
*JC_SNALINKS
#
C41XB01      RLUNAME="C410XB01"
              RDOM="CICS410"
              MODENAME="SMSNA100"
              MAXSESS=8
              MINWIN=6
```

2 Defining the JAM Gateway Configuration

The following table provides descriptions of valid keywords for the `JC_SNALINKS` section:

Keyword	Default	Required/ Optional	Description
<code>MAXSESS</code>	4	Optional	<p>Maximum number of sessions that can be started for a link</p> <p>MAXSESS=nn (4)</p> <p>nn is the maximum number of sessions that can be started for this link. The actual value used is negotiated with the partner and can be lower than this value if the partner is configured with a lower value. This value includes two sessions for the service manager mode. The lowest usable value is 4; it provides two sessions for the application and two sessions for the service manager.</p> <p>Example: <code>MAXSESS=8</code></p> <p>The maximum number of sessions on this link is set to 8 for all modes combined.</p>

Keyword	Default	Required/ Optional	Description
MINWIN	One half the number of MAXSESS	Optional	<p>Minimum number of sessions that will be contention winners</p> <p>MINWIN=nn (MAXSESS/2)</p> <p>nn is the minimum number of sessions that will be contention winners for requests originating from the WebLogic Server side. This keyword defaults to one half the number of MAXSESS, which is suitable in most cases unless asymmetric winners are desired due to application requirements. Also, the default value may not be appropriate if the maximum is negotiated down to a value lower than half of the specified maximum. This value includes one session for the service manager mode. In general, the lowest practical value is 2, which provides one session for the application and one session for the service manager. See IBM's SNA documentation for a complete discussion of session limits and contention winners.</p> <p>Example: MINWIN=6</p> <p>The number of contention winner sessions on this link is set to 6 for all modes combined.</p>
MODENAME	None	Required	<p>Name of mode definition</p> <p>MODENAME=<string></p> <p><string> is the name of a mode definition to be used for applications on this link. The string must match a corresponding mode name defined in the Stack configuration and the VTAM mode table. A valid mode name should be provided by mainframe support personnel.</p> <p>Example: MODENAME="SMSNA100"</p> <p>The CRM will use the SMSNA100 mode for applications on this link.</p>

2 Defining the JAM Gateway Configuration

Keyword	Default	Required/ Optional	Description
RDOM	None	Required	<p>Name of remote domain</p> <p>RDOM=<string></p> <p><string> is a name previously defined as a remote domain on the JC_REMOTE_DOMAINS section of the configuration file. The remote domain name is used as a mechanism for grouping links.</p> <p>Example:</p> <p>RDOM="CRCICS410"</p> <p>This link is associated with the remote domain named CRCICS410.</p>
RLUNAME	None	Required	<p>Alias for remote LU</p> <p>RLUNAME=<string></p> <p><string> is an alias for the remote LU representing the partner application to be used by the CRM.</p> <p>Example:</p> <p>RLUNAME="C410XB01"</p> <p>The CRM routes all traffic for this link to a remote application defined with an alias of C410XB01. This alias must be defined in the stack configuration for a valid mainframe application name.</p>

Keyword	Default	Required/ Optional	Description
SECURITY	LOCAL	Optional	<p>Indicates level of security</p> <p>SECURITY= [LOCAL IDENTIFY VERIFY]</p> <p>One of the specified tokens must be used and the mainframe connection must be configured to match.</p> <p>The meaning of the values are:</p> <ul style="list-style-type: none"> ■ LOCAL = All security is handled by the local system and the link itself has no security requirement. ■ IDENTIFY = A user ID is passed on to the mainframe. This user ID can originate with the client application or it can be a default user ID supplied with the -u option on the gateway startup. ■ VERIFY = A user ID and password are passed on to the mainframe. The user ID can originate with the client application or it can be a default user ID supplied with the -u option on the gateway startup. The password must be supplied by the client application. <p>Refer to the <i>BEA WebLogic Java Adapter for Mainframe Reference Guide</i> for more information about security options.</p> <p>Example:</p> <p>SECURITY=IDENTIFY</p> <p>A user ID is required for all requests made on this link. Note that the mainframe configuration for the remote LU (<i>i.e.</i>, a CICS connection definition) must have a security level that matches.</p>

JC_LOCAL_SERVICES Section

This section of the `jcrmgw.cfg` file maps incoming mainframe program names to a Home interface for an EJB that will service the request. A label identifying the Home interface must precede the first keyword in the set of keywords defining the Home interface.

2 Defining the JAM Gateway Configuration

Listing 2-6 shows an example of the `JC_LOCAL_SERVICES` section. `DPL1SVR` is the name of the program that was invoked from the mainframe and `StatelessSessions.TraderHome` is the name of the Home interface that will be used to invoke the EJB that services this request.

Listing 2-6 Example of `JC_LOCAL_SERVICES` Section

```
*JC_LOCAL_SERVICES
#
StatelessSessions.TraderHome  RNAME="DPL1SVR"
```

The following table provides descriptions of valid keywords for the `JC_LOCAL_SERVICES` section:

Keyword	Default	Required/ Optional	Description
<code>RNAME</code>	None	Required	Name of remote resource associated with a service RNAME=<RRRRRRR> <code>RNAME</code> is the remote resource name associated with this service. For a CICS DPL, this is the actual program name that was invoked from the mainframe. For an IMS request, the resource name must conform to the Transaction ID. A CICS DPL Program is limited to eight characters and an IMS Transaction ID is limited to eight characters. The label on this entry must be the name of a valid home interface for an EJB registered with JNDI. This keyword/value pair is required and has no default value. Example: <code>StatelessSessions.TraderHome</code> <code>RNAME="DPL1SVR"</code>

Keyword	Default	Required/ Optional	Description
SCHEMA	None	Required for use with WebLogic Process Integrator	<p>Identifies the generated DataView.</p> <p>SCHEMA=<string></p> <p>This keyword is used to identify the MFL format used for decoding or encoding data streams.</p> <p>SCHEMA is used by the JAM Plug-in and WebLogic XML/Non-XML Translator when integrating JAM with WebLogic Process Integrator.</p> <p>Example:</p> <p>SCHEMA=myschema</p>

JC_REMOTE_SERVICES Section

This section of the `jcrmgw.cfg` file maps remote mainframe program names to method names that can be used by a local application to invoke the remote request. These remote mainframe program names and associated method names are called remote services. A label identifying the remote service must precede the first keyword in the set of keywords defining the remote service.

[Listing 2-7](#) shows an example of the `JC_REMOTE_SERVICES` section. `DPLINIT` and `TOUPPER` are the labels for the remote services.

Listing 2-7 Example of JC_REMOTE_SERVICES Section

```
*JC_REMOTE_SERVICES
DPLINIT      RDOM="CICS410"
              RNAME="PRIM:DPLINIT"
              TRANTIME=1000
TOUPPER      RDOM="CICS410"
              RNAME="TOUPDPLS"
              TRANTIME=1000
```

The following table provides descriptions of valid keywords for the `JC_REMOTE_SERVICES` section:

2 Defining the JAM Gateway Configuration

Keyword	Default	Required/ Optional	Description
FUNCTION	DPL	Optional	<p>Method of request invocation accepted by the mainframe</p> <p>FUNCTION= [APPC DPL]</p> <p>This keyword indicates the method of request invocation accepted by the mainframe.</p> <ul style="list-style-type: none">■ DPL is only valid when used with a CICS partner and an application that can be invoked using a DPL.■ APPC should be used when invoking services from an IMS server. <p>Example: FUNCTION=APPC</p> <p>This remote service will be invoked as an IMS application rather than a program link.</p>
RDOM	None	Required	<p>Name of remote domain</p> <p>RDOM=<string></p> <p><string> is a name previously defined as a remote domain.</p> <p>Example: RDOM="CRCICS1"</p> <p>This service will be associated with the remote domain named CRCICS1.</p>

Keyword	Default	Required/ Optional	Description
RNAME	None	Required	<p>Remote resource name associated with a service RNAME=< [MMMM:] RRRRRRRR></p> <p>RNAME is the remote resource name associated with this service. For a CICS DPL, this is the actual program name to be invoked. The first portion of this value (MMMM:) is optional and can be an alternate mirror transaction identifier. An alternate mirror transaction is useful for some mainframe database deployments as well as security or charge back systems. The alternate mirror transaction cannot exceed 4 characters in length. The second portion (RRRRRRRR) is the program name to invoke for a CICS DPL. For an APPC style request, the resource name must conform to the Tran ID requirements of IMS.</p> <p>Example: RNAME="AUTH:DPLQRY"</p> <p>This Rname is a program named DPLQRY and will use an alternate mirror transaction name of AUTH.</p>

2 Defining the JAM Gateway Configuration

Keyword	Default	Required/ Optional	Description
SCHEMA or INPUTSCHEMA and/or OUTPUTSCHEMA	None	Optional	<p>Identifies the generated DataView.</p> <p>SCHEMA=<string> INPUTSCHEMA=<string> OUTPUTSCHEMA=<string></p> <p>These keywords are used to identify the MFL format used for decoding or encoding data streams.</p> <p>INPUTSCHEMA and OUTPUTSCHEMA are used by the JAM Plug-in and WebLogic XML/Non-XML Translator when integrating JAM with WebLogic Process Integrator. The INPUTSCHEMA translates the message sent to the mainframe. The OUTPUTSCHEMA translates the message received as a reply from the mainframe.</p> <p>If INPUTSCHEMA and OUTPUTSCHEMA represent the same MFL format, the SCHEMA keyword may be used in place of INPUTSCHEMA and OUTPUTSCHEMA. Do not use SCHEMA with INPUTSCHEMA and/or OUTPUTSCHEMA.</p> <p>The following example uses SCHEMA: SCHEMA=myschema</p> <p>The following example uses both INPUTSCHEMA and OUTPUTSCHEMA: INPUTSCHEMA=myinputschema OUTPUTSCHEMA=myoutputschema</p>
TRANTIME	30000	Optional	<p>Maximum number of milliseconds client will allow for a host request to respond</p> <p>TRANTIME=nnn (30000)</p> <p>nnn is the maximum number of milliseconds the client application will block before a host request is timed out.</p> <p>Example: TRANTIME=120000</p> <p>This remote service will time out if the mainframe does not respond within 2 minutes (120 seconds).</p>

Creating a Configuration File

To create a JAM gateway configuration file for your environment, use any text editor and perform one of the following procedures:

1. Edit the sample `jcrmgw.cfg` file located in the JAM `examples/samples.jar` file with your own configuration requirements.

Or,

2. Create your own configuration file with any text editor.

Note: Refer to the previous section, [“About the JAM Gateway Configuration File”](#) for detailed information about the JAM gateway configuration file parameters.

Verifying the Configuration File with the Configuration Checker Utility

After you create or edit the `jcrmgw.cfg` file, you should verify the contents by invoking the gateway’s Configuration Checker utility directly from a command line. This verification process allows you to discover and correct any errors prior to starting the gateway.

`jcrmConfigurator` is a Java class that is used to check the `jcrmgw.cfg` file. It is recommended that you place the command into a script file and run it with standard output redirected to a file. The resulting output is either diagnostic messages indicating syntax errors in the configuration file, or a formatted listing of the definitions as they are used by the gateway. There are no options for the `jcrmConfigurator` class and the only file that can be processed is the `jcrmgw.cfg` file in the current directory.

To run the `jcrmConfigurator` class:

1. Enter the following commands (assuming `jam.jar` is installed in the `D:\jam42\lib` directory):

2 Defining the JAM Gateway Configuration

```
set classpath=%classpath%;D:\jam42\lib\jam.jar
java com.bea.sna.jcrmgw.jcrmConfigurator
```

2. Review the output and correct any errors that are found.
3. Re-run the `jcrmConfigurator` class and make corrections until all errors are corrected.

What Do I Do Next?

After you create a JAM configuration file for your environment and check it, you can review an example of a mainframe configuration. Refer to [“Putting It All Together”](#) for more information. If you do not wish to review the example, you are ready to learn how to run the CRM so that you can deploy your configuration. Refer to [“Using the CRM Administration Commands”](#) for more information about starting the CRM.

3 Putting It All Together

The topics in this section cover examples of configurations for cross-platform definitions. The configurations put together the information described in [“Preparing Mainframe Configurations for CRM Requirements”](#) and [“Defining the JAM Gateway Configuration.”](#) These examples are for reference only and do not represent all configuration possibilities. Consult with your system administrator for specific information about your system.

Action List

Before you review the sample configurations included in this section, see the following action list and refer to the appropriate information sources.

	Your action...	Refer to...
1	Complete all prerequisite tasks.	“Prerequisites”
2	Review an example of Windows NT or UNIX cross-platform definitions.	“Example of Windows NT or UNIX Cross-Platform Definitions”
3	Review an example of OS/390 cross-platform definitions.	“Example of OS/390 Definitions”
4	Prepare for the next step.	“What Do I Do Next?” and “Using the CRM Administration Commands”

Prerequisites

Before you use the CRM Administrative commands, you should complete the following tasks:

Your action...	Refer to...
1 Verify that system requirements have been met.	<i>BEA WebLogic Java Adapter for Mainframe Introduction and Release Notes</i>
2 Determine which configuration your system requires and install the JAM product as appropriate for your configuration.	<i>BEA WebLogic Java Adapter for Mainframe Installation Guide</i>
3 Learn about mainframe configuration requirements for the CRM.	“Preparing Mainframe Configurations for CRM Requirements”
4 You may have already created a JAM configuration file and verified that the gateway was properly configured. This action may be done before or after reviewing the configuration examples.	“Defining the JAM Gateway Configuration”

Example of Windows NT or UNIX Cross-Platform Definitions

Before installing JAM software, review [Table 3-1](#) for an example of Windows NT or UNIX cross-platform definitions. Consult with your VTAM system administrator to obtain the value indicated in the *Name* column and make the corresponding entries shown in the *Needed In* column.

Note: DCL-based stacks referred to in [Table 3-1](#) include the following stacks:

- Solaris SNAP-IX
- IBM Comm Server for Windows NT

Listing 3-1 shows the corresponding JAM configuration file.

Table 3-1 Summary of Windows NT or UNIX Definitions

Name	Originates In	Needed In
SNA Network ID (e.g. SNANET1) and VTAM Host ID (e.g. VTAMHOST)	VTAM configuration	DCL-based Stack Configuration: Example: fqcp_name= SNANET1 .SPARC1 adj_cp_name= SNANET1 .VTAMHOST fqplu_name= SNANET1 .CICSSYN
Mode Name (e.g. SNA62)	VTAM-MODEENT definition	CICS Sessions Definition: Example: MODENAME (SNA62) DCL-based Stack Configuration: Example: mode_name= SNA62 JCRMGW Configuration: Example: JC_SNALINKS MODENAME=" SNA62 "
Control Point Name CPNAME (e.g. SPARC1)	VTAM-PU definition	DCL-based Stack Configuration: Example: fqcp_name=SNANET1.SPARC1 cp_alias= SPARC1
Local LU Name (e.g. L0F0024A)	VTAM-LU definition	CICS CONNECTION definition: Example: NETNAME (L0F0024A) DCL-based Stack Configuration: Example: lu_name= L0F0024A lu_alias= L0F0024A JCRMGW Configuration: Example: JC_SNASTACKS LOCALLU=" L0F0024A "

3 Putting It All Together

Table 3-1 Summary of Windows NT or UNIX Definitions

Name	Originates In	Needed In
CICS LU Name (e.g. CICSSYN)	VTAM-LU definition	DCL-based Stack Configuration: Example: fqplu_name=SNANET1.CICSSYN plu_alias=CICSSYN JCRMGW Configuration: Example: JC_SNALINKS RLUNAME="CICSSYN"
Terminal Identifier (e.g. 05DF0024)	VTAM (IDNUM+IDBLK)	DCL-based Stack Configuration: Example: node_id=<05000002>
SYNCLVL supported by jcrmgw	Stacks	DCL-based Stack Configuration: Example: [define local_lu] syncpt_support=CONFIRM
Map all incoming conversations to JAM gateway (make sure TPs have all privileges available, e.g. CNOS, service conversations, etc.).	Stacks	DCL-based Stack Configuration: Example: Sna_tps <404040...hex representation of 64 EBCDIC spaces...404040> TYPE=QUEUED TIMEOUT=-1 USERID=authorized_user_here GROUP=authorized_group_here LUALIAS=LOF0024A JCRMGW Configuration: Example: JC_LOCAL_SERVICES RNAME="DPL1SVR"
CICS DPL program name (e.g. TOUPPER)	CICS/ESA	JCRMGW Configuration: Example: JC_REMOTE_SERVICES RNAME=TOUPPER

Listing 3-1 Corresponding jcrmgw.cfg Configuration File

```
*JC_REMOTE_DOMAINS
#
CICS13          DOMAINID="13"
#
*JC_SNACRM
#
CRMAN  SNACRMADDR="//da1nt66:8650"
          GROUP="G1"

*JC_SNASTACKS
#
OS390          STACKTYPE="SPX62"
          LOCALLU="LOF0024A"
#
*JC_SNALINKS
#
CICS          RLUNAME="CICSSYN"
          RDOM="CICS13"
          MODENAME="SNA62"
          MAXSESS=8
          MINWIN=6
#
*JC_LOCAL_SERVICES
#
TraderHome     RNAME="DPL1SVR"
#
*JC_REMOTE_SERVICES
TOUPPER        RDOM="CICS13"
          RNAME="TOUPPER"
          TRANTIME=10000
```

Example of OS/390 Definitions

Before installing JAM software, review [Table 3-2](#) for an example of SNA definitions when the CRM runs on an OS/390 platform. Consult with your system administrator to obtain the value indicated in the *Name* column and make the corresponding entries shown in the *Needed In* column. [Listing 3-2](#) shows the corresponding JAM configuration file.

Table 3-2 Summary of OS/390 SNA Definitions

Name	Originates In	Needed In
Local LU Name (e.g. BEAAPPL1)	VTAM-LU definition	<p>CICS CONNECTION definition: Example: NETNAME (BEAAPPL1)</p> <p>VTAM Configuration: Example: BEASNA VBUILD TYPE=APPL BEAAPPL1 APPL ACB=BEAAPPL1, APPC=YES, PARSESS=YES</p>
Mode Name (e.g. SNA62)	VTAM-MODEENT definition	<p>CICS Sessions Definition: Example: MODENAME (SNA62)</p> <p>JCRMGW Configuration: Example: JC_SNALINKS MODENAME= "SNA62"</p> <p>VTAM Configuration (not required): Example: MODEENT=SNA62</p>
CICS LU Name (e.g. CICSSYN)	VTAM-LU definition	<p>JCRMGW Configuration: Example: JC_SNALINKS RLUNAME= "CICSSYN"</p>
Maximum sync-level allowed by jcrmgw	jcrmgw	<p>VTAM Configuration: Example: SYNCLVL=CONFIRM</p>

Table 3-2 Summary of OS/390 SNA Definitions

Name	Originates In	Needed In
CICS DPL program name (e.g. TOUPPER)	CICS/ESA	JCRMGW Configuration: Example: JC_REMOTE_SERVICES RNAME=TOUPPER

Listing 3-2 Corresponding jcrmgw.cfg Configuration File

```

*JC_REMOTE_DOMAINS
#
CICS13          DOMAINID="13"
#
*JC_SNACRM
#
CRMAN  SNACRMADDR="//da1nt66:8650"
        GROUP="G1"

*JC_SNASTACKS
#
OS390          STACKTYPE="SPX62"
               LOCALLU="BEAAPPL1"
#
*JC_SNALINKS
#
CICS          RLUNAME="CICSSYN"
               RDOM="CICS13"
               MODENAME="SNA62"
               MAXSESS=8
               MINWIN=6
#
*JC_LOCAL_SERVICES
#
TraderHome    RNAME="DPL1SVR"
#
*JC_REMOTE_SERVICES
TOUPPER      RDOM="CICS13"
               RNAME="TOUPPER"
               TRANTIME=10000

```

What Do I Do Next?

After you create a JAM configuration file for your environment and check it, you are ready to learn how to run the CRM so that you can deploy your configuration. Refer to [“Using the CRM Administration Commands”](#) for more information about starting the CRM.

4 Using the CRM Administration Commands

The topics in this section cover commands an administrator uses to start and stop the Communications Resource Manager (CRM). These commands may be entered on the command line for UNIX or NT systems or through Job Control Language (JCL) on OS/390MVS.

Action List

Before you use the CRM Administrative commands, see the following action list and refer to the appropriate information sources.

Your action...	Refer to...
1 Complete all prerequisite tasks.	“Prerequisites”
2 Learn how to start or stop the CRM.	“Starting and Stopping the CRM”
3 Prepare for the next step.	“What Do I Do Next?” and “Deploying Your Configuration”

Prerequisites

Before you use the CRM Administrative commands, you should complete the following tasks:

	Your action...	Refer to...
1	Verify that system requirements have been met.	<i>BEA WebLogic Java Adapter for Mainframe Introduction and Release Notes</i>
2	Determine which configuration your system requires and install the JAM product as appropriate for your configuration.	<i>BEA WebLogic Java Adapter for Mainframe Installation Guide</i>
3	Verify that your mainframe configuration has been properly set up for the CRM.	“Preparing Mainframe Configurations for CRM Requirements”
4	Create a JAM configuration file and verify that the gateway has been properly configured.	“Defining the JAM Gateway Configuration”

Starting and Stopping the CRM

Before you learn how to start or stop the CRM for your specific operating system, you need to understand the start and stop command syntax and options. The two commands for starting and stopping the CRM are:

- [“CRM Command”](#) to launch the CRM
- [“crmdown Command”](#) to shut the CRM down

These commands can be entered from a command line or you may also use JCL on an OS/390 Multiple Virtual Storage (MVS) platform to set the invoke the CRM start and stop commands. Sample JCL is included in the `samples.jar` file, which is delivered with your JAM installation.

The following sections provide detailed descriptions of these commands and examples of JCL for each command. After you learn about the commands, you can then refer to the specific instructions for using them with your operating system in the sections following the command descriptions.

CRM Command

The `CRM` command launches the Communications Resource Manager. The CRM communicates between a JAM gateway and a remote application using SNA protocol.

When you start the CRM from the command line, the CRM Command Line Console puts its prompt in the window. If you exit the Command Line Console window, all of the active links are shut down.

You must configure one CRM for each JAM gateway, as well as configuring one stack for each CRM definition. Each stack can manage one or more SNA links. The CRM must be started and listening before the JAM gateway is booted.

The CRM has a `BLOBLOG` log file stored in `$APPDIR`. `BLOBLOG` stores session and link information. You can use the `CRMLOGS` command to display the contents and state of the CRM log file.

Example of the `CRM` command line:

```
CRM [ -t 0|1|2|3 ] [-s] [-p <nbr>] <addr> <group>
```

Descriptions of the command line options follow.

Command Line Options

The following table provides descriptions of the valid options for the `CRM` command:

4 Using the CRM Administration Commands

Keyword	Default	Required/ Optional	Description
-t [0 1 2 3]	0	Optional	<p>Turns tracing on and indicates the level of tracing.</p> <p>0=No tracing.</p> <p>Setting this level effectively disables CRM tracing and closes the trace file, if there is one. If tracing is subsequently restarted, a new file is created with an incremental numerical suffix. 0 is the default trace level.</p> <p>1=Minimum tracing.</p> <p>At this level, the CRM traces only major events and is sufficient only to determine the sequence of application conversations.</p> <p>2=Medium tracing.</p> <p>At this level, the CRM also traces all I/O buffers.</p> <p>3=Maximum tracing.</p> <p>At this level, the CRM also traces all APPC verbs.</p> <p>Note: Trace options for the CRM may also be set from the CRM Monitor. Refer to “Using the CRM Monitor” for more information about setting trace options from the CRM Monitor.</p>

Keyword	Default	Required/ Optional	Description
-s	Off (if parameter is not used)	Optional	<p>Indicates APPC Stack API trace is turned on. On OS/390 platforms, the General Trace Facility (GTF) is used to capture API trace records under user EID 2EA. The GTF must be active on the OS/390 platform to use the -s parameter.</p> <p>If the APPC Protocol Stack API trace is enabled, it generally shows the parameters and results of all API calls. Depending on the stack being used, other options (such as vendor-specified environment variables) may have to be activated for the CRM to enable the trace.</p> <p>Note: Trace options for the APPC Stack API trace may also be set from the CRM Monitor. Refer to “Using the CRM Monitor” for more information about setting trace options from the CRM Monitor.</p>
-p <nbr>	100 threads	Optional	<p>Turns on the performance option and indicates the number of threads to start.</p> <p>This value should correspond to the load of SNA requests that will be made concurrently. If the number of requests exceeds the number of threads, the request is still executed; however, the completion time could be affected. Do not exceed 200 threads. The CRM is tuned for a maximum of 200 threads. Lower the threads value if you have a restriction on the number of threads that can be active in your system.</p>

4 Using the CRM Administration Commands

Keyword	Default	Required/ Optional	Description
<addr>	None	Required	Specifies a TCP/IP address using //hostname:port_addr or the sockaddr_in format of family, port, address: <0xFFFFPPPPAAAAAAA> In this entry, arguments and options are defined in the following way: FFFF is the hex value of the protocol family, always 0x0002 for the INET family. PPPP is the hex value of an unused TCP/IP port. AAAAAAA is the hex value of the IP address for the machine running the CRM.
<group>	None	Required	Indicates the JAM Gateway Group Name.

Environment Variables

You must set the following environment variables before starting the CRM:

- From the UNIX command line:

APPDIR must be set to the application directory.

For UNIX, the APPDIR indicates the directory for the files created by the CRM.

- From MVS:

APPDIR=<High level qualifier for datasets to be created in APPDIR>

For MVS, the APPDIR qualifies the dataset for the files created by the CRM.

Diagnostics

The CRM exits with a return code of 0 upon successful completion.

Examples

To launch the CRM with the console running in the background, enter the following command:


```
$ CRM //myhost:5587 GROUP2 std.out 2>std.err &
```

To launch the CRM with detailed tracing and APPC Stack API tracing turned on from the command line using the host/port address, enter the following command:

```
CRM -t 3 -s //myhost:5587 GROUP2
```

Sample CRM Command JCL for OS/390 MVS Platform

This section provides an explanation of the `SET` commands and an example of JCL that can be used when you run the `CRM` command.

- `SET STARTCMD` Sets the `CRM` command line parameters.
- `SET OBJLIB` Indicates the name of the PDSE library where the `CRM` executable is installed.
- `SET DATA` Indicates the dataset containing the `ENVFILE`.
- `SET ENVFILE` Indicates the name of the PDS member that contains the environment variables for the `CRM`. A sample member, `ENV`, is delivered with your product.
- `SET SIZE` Defines the region size for the running `CRM` task. The recommended setting for this option is `0M` to allow the `CRM` to start up and level out to the size it requires.
- `SET ENV` Indicates the `ENVFILE` DD name. This value is pre-set and should not be changed.
- `SET CEE` Specifies the high-level qualifier for the LE runtime library. `CEE` should be set to the prefix of the IBM Language Environment data sets. Language Environment is required to run the `CRM`.
- Note:** Uncomment the `SET CEE` line and tailor the `STEPLIB` concatenation if these libraries are not in your system link library concatenation.
- `SET CBC` Specifies the high-level qualifier for the C/C++ runtime library. `CBC` should be set to the prefix of the IBM C++ data sets.

Listing 4-1 Sample `crmstart.jcl` for `CRM` Command

```
//*****  
**  
//* THIS JOB IS USED TO RUN THE SNACRM PROCESS. *  
//* * *  
//* @(#) $Id: crmstart.jcl,v 1.3 2001/05/07 23:41:27 crout Exp $ *  
//* Copyright (c) 2000 BEA Systems, Inc., all rights reserved. *
```

4 Using the CRM Administration Commands

```
//*****
**
//* YOU MUST SET THE ENVIRONMENT VARIABLES NEEDED BY SNACRM *
//*****
**
//*****
**
//* USE THE SET STATEMENTS TO SET THE APPROPRIATE VALUES *
//* STARTCMD IS THE CRM COMMAND LINE *
//* OBJLIB IS THE LOAD LIBRARY CONTAINING THE PROGRAM EXECUTABLES*
//* DATA IS THE DATASET THAT CONTAINS THE ENVIRONMENT VARIABLES *
//* ENVFILE NAMES THE MEMBER THAT CONTAINS THE ENVIRONMENT VARS *
//* RUNOPTS SETS ANY DESIRED LE RUNTIME OPTIONS (OPTIONAL) *
//* SIZE SETS THE REGION SIZE FOR THE SNACRM PROCESS. 0M SETS NO *
//* LIMITS ON THE REGION SIZE *
//* TAILOR YOUR JCL FOR THE BELOW IF THESE LIBRARIES ARE NOT *
//* IN YOUR SYSTEM LINK LOAD LIBRARY CONCATENATION *
//* CEE IS THE HLQ FOR THE LE RUNTIME LIBRARY *
//* CBC IS THE HLQ FOR THE C/C++ RUNTIME LIBRARY *
//*****
**
// SET STARTCMD=' " //<address>:<port>" <group>'
// SET OBJLIB=
// SET DATA=
// SET ENVFILE=ENV
// SET RUNOPTS=
// SET SIZE=0M
// SET ENV=' ENVAR (" _CEE_ ENVFILE=DD:ENV" ) '
//* SET CEE=CEE, CBC=CBC
//CRM EXEC PGM=CRM, REGION=&SIZE,
// PARM=' POSIX(ON) &ENV &RUNOPTS/&STARTCMD '
//STEPLIB DD DSN=&OBJLIB, DISP=SHR
//* DD DSN=&CEE..SCEERUN, DISP=SHR
//* DD DSN=&CBC..SCLBDLL, DISP=SHR
//MSGFILE DD SYSOUT=*
//TRACE DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//ENV DD DSN=&DATA (&ENVFILE) , DISP=SHR
//
```

crmdown Command

The `crmdown` command shuts down a CRM.

`crmdown` shuts down the CRM specified on the command line. `crmdown` can be used from any machine located on the same TCP/IP network as the machine running the CRM server. The command can also be used in a script and it returns 0 if the command could be sent to the target CRM or 1 if the command could not be sent to the target CRM.

Example of the `crmdown` command line:

```
crmdown -n<hostname:port> [-v -i]
```

Descriptions of the command line options follow.

Command Line Options

The following table provides descriptions of the valid options for the `crmdown` command:

Keyword	Default	Required/ Optional	Description
<code>-n<hostname:port></code>	None	Required	Names the machine and port running the CRM server.
<code>-v</code>	Off	Optional	Specifies verbose. Normally the command will not produce any messages, facilitating use in a script
<code>-i</code>	Off	Optional	Ignores errors.

Diagnostics

`crmdown` only checks the syntax of the command. If the command could not be successfully sent to the CRM, `crmdown` prints an error message if in verbose mode and exits with error code 1. Upon successful completion, `crmdown` exits with exit code 0.

Example

To stop the CRM running on `mach1` at port 5000:

```
crmdown -n mach1:5000
```

Sample crmdown Command JCL for OS/390 MVS Platform

This section provides an explanation of the SET commands and an example of JCL that can be used when you run the `crmdown` command.

- SET STOPCMD Sets the `crmdown` command line parameters.
- SET OBJLIB Indicates the name of the PDSE library where the `crmdown` executable is installed.
- SET DATA Indicates the dataset containing the `ENVFILE`.
- SET ENVFILE Indicates the name of the PDS member that contains the environment variables for the `crmdown`. A sample member, `ENV`, is delivered with your product.
- SET SIZE Defines the region size for the running `crmdown` task.
- SET ENV Indicates the `ENVFILE DD` name.
- SET CEE Specifies the high-level qualifier for the LE runtime library. CEE should be set to the prefix of the IBM Language Environment data sets. Language Environment is required to run `crmdown`.
- SET CBC Specifies the high-level qualifier for the C/C++ runtime library. CBC should be set to the prefix of the IBM C++ data sets.

Note: Uncomment the SET CBC line and tailor the STEPLIB concatenation if these libraries are not in your system link library concatenation.

Listing 4-2 Sample crmdown.jcl for crmdown Command

```
//*****  
**  
//* THIS JOB IS USED FOR THE STAND-ALONE COMMAND USED *  
//* TO SHUTDOWN THE SNACRM PROCESS. SEE USER GUIDE FOR MORE INFO *  
// *  
//* @(#) $Id: crmdown.jcl,v 1.5 2001/05/07 23:41:27 crout Exp $ *  
//* Copyright (c)2000 BEA Systems, Inc., all rights reserved. *  
//*****  
**  
// * YOU MUST SET THE ENVIRONMENT VARIABLES NEEDED BY CRMDOWN *  
//*****  
**  
//*****  
**  
// * STOPCMD INDICATES THE COMMAND LINE FOR CRMDOWN *
```

```
/** OBJLIB IS THE LOAD LIBRARY CONTAINING THE PROGRAM EXECUTABLES*
/** RUNOPTS SETS ANY DESIRED LE RUNTIME OPTIONS (OPTIONAL) *
/** DATA IS THE DATASET THAT CONTAINS THE ENVIRONMENT VARIABLES *
/** ENVFILE NAMES THE MEMBER THAT CONTAINS THE ENVIRONMENT VARS *
/** SIZE SETS THE REGION SIZE FOR THE SNACRM PROCESS *
/** *
/** TAILOR YOUR JCL FOR THE BELOW IF THESE LIBRARIES ARE NOT *
/** IN YOUR SYSTEM LINK LOAD LIBRARY CONCATENATION *
/** CEE IS THE HLQ FOR THE LE RUNTIME LIBRARY *
/** CBC IS THE HLQ FOR THE C/C++ RUNTIME LIBRARY *
/*******
**
// SET STOPCMD='-n<host name>:<port>'
// SET OBJLIB=
// SET RUNOPTS=
// SET DATA=
// SET ENVFILE=ENV
// SET SIZE=1M
// SET ENV='ENVAR("_CEE_ENVFILE=DD:ENV")'
/** SET CEE=CEE,CBC=CBC
//CRMDOWN EXEC PGM=CRMDOWN,REGION=&SIZE,
// PARM=' POSIX(ON) &ENV &RUNOPTS/&STOPCMD'
//STEPLIB DD DSN=&OBJLIB,DISP=SHR
/** DD DSN=&CEE..SCEERUN,DISP=SHR
/** DD DSN=&CBC..SCLBDLL,DISP=SHR
//ENV DD DSN=&DATA(&ENVFILE),DISP=SHR
//MSGFILE DD SYSOUT=*
//SYSPRINT DD SYSOUT=*

//
```

Starting the CRM on UNIX, OS/390 UNIX, or Windows NT

The CRM server communicates directly with the PU 2.1 server to provide SNA connectivity. These servers can be started manually. The PU 2.1 server must always be started before the CRM. Both servers must be started before starting the associated JAM gateway.

To ensure proper start up of the CRM, complete the following tasks:

1. Set the following environment variable in the environment where the CRM is started:

4 Using the CRM Administration Commands

`APPDIR` must be set to the application directory where the CRM will create its files.

2. Start the PU2.1 Server.

Refer to the operational documentation provided by your SNA stack vendor for information about starting the PU2.1 server. The SNA stack must be running and active before you start the CRM.

3. Start the CRM.

- To start the CRM on UNIX:

Enter the `CRM` command with your required option arguments on the command line.

When you start the CRM from the UNIX command line, the CRM Command Line Console puts its prompt in a window, and if exited, shuts down all of the active links.

- To start the CRM from a Command Prompt Window on Windows NT:

Open a Command Prompt window and navigate to the drive where the CRM executable is installed. Enter the `CRM` command with your required option arguments to start `CRM.exe`.

On Windows NT, the CRM can be started in its own window; however, no prompt is displayed and no console commands are available.

Refer to [“CRM Command”](#) for more detailed information about the `CRM` command.

Starting the CRM on OS/390 MVS

The OS/390 MVS platform sets the environment and invokes the CRM using Job Control Language (JCL).

1. Set the following environment variable in the environment where the CRM is started. A sample file is delivered (`ENV`) in the data library.

```
APPDIR=<High level qualifier for datasets to be created in  
APPDIR>
```

The high level qualifier is used for the datasets created by the CRM and must be unique for each running CRM.

Note: For more information about the dataset names, refer to the information for installing the CRM on an MVS platform in the *BEA WebLogic Java Adapter for Mainframe Installation Guide*.

2. Run a CRM job using JCL written for your system.

Refer to “[Sample CRM Command JCL for OS/390 MVS Platform](#)” for more information about the sample `CRMstart.JCL`.

Note: If the CRM is installed on an OS/390 MVS platform, it does not have to be restarted when the JAM gateway shuts down abnormally. The CRM continues running until it receives a normal termination command. Only the OS/390 MVS version and the OS/390 UNIX version of the CRM have this persistent feature. Refer to “[Stopping the CRM On All Platforms](#)” for more information about stopping the CRM on an OS/390 MVS platform.

Stopping the CRM On All Platforms

If the JAM gateway terminates, the persistent CRM feature allows the OS/390 CRM to continue running. You must explicitly shutdown the CRM if the JAM gateway has terminated and the CRM needs to be stopped.

■ To stop the CRM from a UNIX or Windows NT system:

Enter the `crmdown` command with your required option arguments on the command line.

■ To stop the CRM from OS/390 MVS:

Run a `crmdown` job using JCL written explicitly for your system.

Refer to “[crmdown Command](#)” for more information about the `crmdown` command and the sample `crmdown JCL`.

What Do I Do Next?

After you learn how to start and stop the CRM , you are ready to deploy your configuration file and verify that the JAM System is running correctly. Refer to [“Deploying Your Configuration”](#) for more information.

5 Deploying Your Configuration

After you create your JAM gateway configuration file, you need to deploy it, set the WebLogic Server CLASSPATH, and add your JAM startup class to the WebLogic Server configuration file. Then you can verify that your JAM system is running. The topics in this section cover the steps you need to complete to start your JAM system.

Action List

Before you deploy your configuration and start the JAM system, see the following action list and refer to the appropriate information sources.

Your action...	Refer to...
1 Complete all prerequisite tasks.	“Prerequisites”
2 Deploy the JAM gateway configuration file.	“Placing the Gateway Configuration File in the WebLogic Server Start Directory”
3 Set your WebLogic Server CLASSPATH to locate the JAM .JAR files.	“Setting the WebLogic Server CLASSPATH for the JAM.JAR File”
4 Add the JAM startup class to the WebLogic Server configuration file.	“Adding the JAM Startup Class to the WebLogic Server Configuration File”
5 Verify that the JAM gateway is working correctly.	“Verifying that the JAM Gateway is Ready”

Your action...	Refer to...
6 Prepare for the next step.	“What Do I Do Next?” and “Using JAM Administration Utilities”

Prerequisites

Before you deploy your configuration and start the JAM system, you should complete the following tasks:

Your action...	Refer to...
1 Verify that system requirements have been met.	<i>BEA WebLogic Java Adapter for Mainframe Introduction and Release Notes</i>
2 Determine which configuration your system requires and install the JAM product as appropriate for your configuration.	<i>BEA WebLogic Java Adapter for Mainframe Installation Guide</i>
3 Verify that your mainframe configuration has been properly set up for the CRM.	“Preparing Mainframe Configurations for CRM Requirements”
4 Create a JAM configuration file and verify that the gateway has been properly configured.	“Defining the JAM Gateway Configuration”
5 Learn how to start or stop the CRM with the CRM administration commands.	“Using the CRM Administration Commands”

Placing the Gateway Configuration File in the WebLogic Server Start Directory

WebLogic Server supports multiple configured server domains.

Place the `jcrmgw.cfg` file in the top-level WebLogic directory similar to the following example:

```
<bea_home>/<install_dir>/config/<domain_name>
```

In this example, the directory resolves to the resources for one started instance of a WebLogic Server.

Setting the WebLogic Server CLASSPATH for the JAM.JAR File

When you installed your JAM product, you should have already set your WebLogic Server CLASSPATH as described in the *BEA WebLogic Java Adapter for Mainframe Installation Guide*.

If you have not already done so, use the following command to place the `jam.jar` file in your CLASSPATH:

```
<JAM 42 installation directory>/lib/jam.jar
```

Adding the JAM Startup Class to the WebLogic Server Configuration File

The JAM startup class, `com.bea.jcrmgw.gwboot`, must be added to the WebLogic Server configuration file called `config.xml` to start the JAM gateway. The JAM gateway is then automatically started when you start WebLogic Server. The startup class can be added through the WebLogic Server Administration Console. Refer to the *BEA WebLogic Server Administration Guide* for specific instructions on adding a Startup Class to the configuration using the WebLogic Server Administration Console.

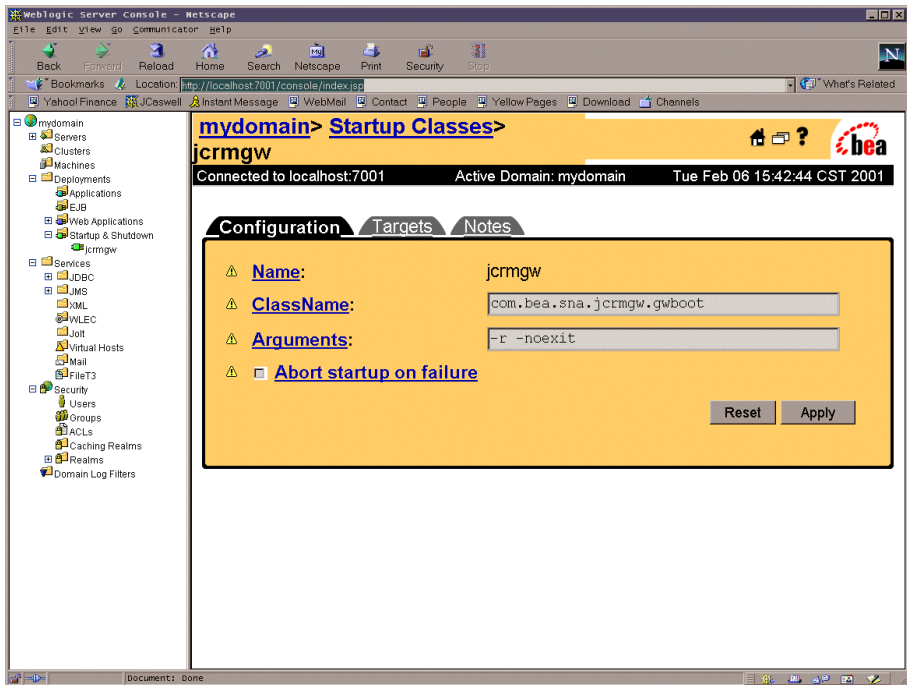
Note: Be sure to set the CLASSPATH as described in “[Setting the WebLogic Server CLASSPATH for the JAM.JAR File](#)” before you start WebLogic Server.

[Figure 5-1](#) shows an example of the WebLogic Server Administration Console displaying the settings for a JAM gateway that communicates with a distributed CRM that is started in a remote environment. The console indicates that WebLogic Server will not stop if the JAM gateway fails to start because the **Abort startup on failure** option is not checked.

Refer to the following general tips for adding the JAM startup class through the WebLogic Server Administration Console.

- Click the `jcrmgw` link to display detail about the class as shown in [Figure 5-1](#).

Figure 5-1 Details of the jcrmgw Startup Class



- Note that the `-r` and `-noexit` arguments are used.

The following arguments can be used with JAM:

Argument	Description
<code>-r</code>	Specifies a remote CRM and suppresses the spawning of a CRM process. The CRM is assumed to already be running and listening at the address specified in the <code>CRMADDR</code> entry of the <code>jcrmgw.cfg</code> file. This argument is required when the CRM is running on a different machine from the JAM gateway, or if the CRM was started in its own window for purposes of testing a configuration.
<code>-t</code>	Specifies tracing CRM. Turns on level 3 CRM tracing. This option can only be used if the gateway is spawning a new local CRM. If the CRM is running remotely and tracing is required, use the <code>-t</code> option on the command line that started the CRM.

Argument	Description
-noexit	Prevents the WebLogic Server from exiting when the gateway shuts down.

- Do not select the `Abort startup on failure` option so that WebLogic Server will not stop if the JAM gateway fails to be brought up.

Verifying that the JAM Gateway is Ready

After you deploy the configuration file, set the WebLogic Server CLASSPATH, and add your JAM startup class to the WebLogic Server configuration file, you are ready to verify that the gateway is ready.

Then, you can verify that your JAM gateway is ready by performing the following checks:

- Start the CRM. Refer to [“Starting and Stopping the CRM”](#) for detailed information about starting the CRM as appropriate for your operating system.
- Re-start WebLogic Server, which also starts the JAM gateway. When JAM has successfully booted and is ready for clients, the following message is given on the WebLogic console and in the WebLogic log files:

```
<(Date) (Time)> <Notice> <jcrmgw> <JAM gateway ready for use>
```
- Use the CRM Monitor to verify the CICS connection. Refer to the next section, [“Using the CRM Monitor”](#) for detailed information about the CRM Monitor.

What Do I Do Next?

After you deploy your JAM configuration and start the system, you are ready to learn how to use the JAM Administration utilities, the CRM Monitor and JAM Administration Servlet. You will use the CRM Monitor to verify that your JAM gateway is running properly. Refer to [“Using JAM Administration Utilities”](#) for more information.

6 Using JAM Administration Utilities

The BEA WebLogic Java Adapter for Mainframe (JAM) product includes the following administration utilities for managing the Communications Resource Manager (CRM) and the JAM gateway:

- Administrative commands allow you to activate and deactivate links.
- The CRM Monitor allows you to monitor the CRM server that provides SNA connectivity, select trace options, and monitor link status.
- The JAM Administration Servlet allows you to start or stop the gateway when the system requires administration or maintenance and you do not want to stop or restart your WebLogic Server.

The CRM Monitor and JAM Administration Servlet are used when the CRM is running on Windows or UNIX operating systems. If your CRM is running on an OS/390 UNIX operating system, you must enter administrative commands from the command line.

Action List

Before you use the JAM administration utilities, see the following action list and refer to the appropriate information sources.

	Your action...	Refer to...
1	Complete all prerequisite tasks.	“Prerequisites”
2	Learn how to activate and de-activate links with administrative commands.	“Activating and De-Activating Links”
3	Learn about the CRM Monitor.	“Using the CRM Monitor”
4	Start the CRM Monitor as appropriate for your operating system.	“Launching the CRM Monitor from the Windows NT Desktop” and “Launching the CRM Monitor from the Command Line”
5	Set the CRM Monitor options.	“Setting CRM Monitor Options”
6	Learn about the JAM Administration Servlet.	“Starting and Stopping the Gateway with the JAM Administration Servlet”
7	Start the JAM Administration Servlet.	“Launching the JAM Administration Servlet”
8	Set the JAM Administration Servlet options.	“Setting JAM Administration Servlet Options”
9	Start or stop the JAM gateway using the JAM Administration Servlet.	“Using JAM Administration Servlet Start and Stop Commands”
10	Determine what you want to do next.	“What Do I Do Next?,” “Shutting Down the System for Planned Outages,” <i>BEA WebLogic Java Adapter for Mainframe Workflow Processing Guide</i> , or <i>BEA WebLogic Java Adapter for Mainframe Programming Guide</i>

Prerequisites

Before you use the JAM administration utilities, you should complete the following tasks:

Your action...	Refer to...
1 Verify that system requirements have been met.	<i>BEA WebLogic Java Adapter for Mainframe Introduction and Release Notes</i>
2 Determine which configuration your system requires and install the JAM product as appropriate for your configuration.	<i>BEA WebLogic Java Adapter for Mainframe Installation Guide</i>
3 Verify that your mainframe configuration has been properly set up for the CRM.	“Preparing Mainframe Configurations for CRM Requirements”
4 Create a JAM configuration file and verify that the gateway has been properly configured.	“Defining the JAM Gateway Configuration”
5 Learn how to start or stop the CRM with the CRM administration commands.	“Using the CRM Administration Commands”
6 Deploy your configuration and verify that the gateway is working properly.	“Deploying Your Configuration”

Activating and De-Activating Links

You can activate and de-activate CRM links that have been defined in the `JC_SNALINKS` section of the `jcrmgw.cfg` file by executing one of the link commands from the command line. There are two commands used to activate and de-activate links:

- [“crmlkon Command”](#)
- [“Sample crmlkon Command JCL for OS/390 MVS Platform”](#)

You may also use Job Control Language (JCL) on an OS/390 Multiple Virtual Storage (MVS) platform to set the environment and invoke link commands. The following sections provide descriptions of the link commands and samples of JCL that may be used for your MVS operating system.

crmlkon Command

The `crmlkon` command starts one or more named CRM links.

`crmlkon` starts all of the CRM links named on the command line. This command is useful if one or more individual links failed to start when the CRM server booted. It can be used from any machine located on the same TCP/IP network as the machine running the CRM server. It can be used in a script and returns 0 if the command could be sent to the target CRM. It returns 1 if the command could not be sent to the target CRM.

Example of the `crmlkon` command line:

```
crmlkon -n<hostname:port> [-v -i] <linkname> ...
```

Descriptions of the command line options follow.

Command Line Options

The following table provides descriptions of the valid options for the `crmlkon` command:

Keyword	Default	Required/ Optional	Description
-n<hostname:port>	None	Required	Names the machine and port running the CRM server.
-v	Off	Optional	Specifies verbose. Normally the command will not produce any messages, facilitating use in a script.

Keyword	Default	Required/ Optional	Description
-i	Off	Optional	Ignores errors. When specifying multiple links, any error encountered when issuing CRM commands causes <code>crmlkon</code> to stop processing links and return. Errors can be ignored for individual links and processing continues with the next named link
<linkname>	None	Required	Names the link to be started. This is the <code>JC_SNALINKS</code> entry in the <code>jcrmgw.cfg</code> that defines this link. Multiple link names can be specified.

Example

To start links, `link2` and `cicstest`, owned by the CRM running on `mach1` at port 5000:

```
crmlkon -n mach1:5000 link2 cicstest
```

Diagnostics

`crmlkon` only checks the syntax of the command. Use the CRM Monitor to determine if the link actually became active. Refer to [“Setting CRM Monitor Options”](#) for more information. If the command could not be successfully sent to the CRM, `crmlkon` prints an error message if in verbose mode and exits with error code 1. Upon successful completion, `crmlkon` exits with exit code 0.

Sample crmlkon Command JCL for OS/390 MVS Platform

This section provides an explanation of the `SET` commands and an example of JCL that can be used when you run the `crmlkon` command. The sample `SET` commands may not reflect the configuration of your system. You must customize the `SET` commands for your environment. Refer to your System Administrator for more information about your particular setup.

- SET LINKCMD Sets the `crmlkon` command line parameters. Refer to “[crmlkon Command](#)” for more information about the command line parameters.
- SET OBJLIB Indicates the name of the PDSE library where the `crmlkon` executable is installed.
- SET DATA1 Indicates the name of the PDS library where the `crmlkon` required parameter file FMB was installed.
- SET DATA Indicates the dataset containing the `ENVFILE`.
- SET ENVFILE Indicates the name of the PDS member that contains the environment variables for the `CRMLKON`. A sample member, `ENV`, is delivered with your product.
- SET SIZE Defines the region size for the running `crmlkon` task.
- SET ENV Indicates the `ENVFILE` DD name. This value is pre-set.
- SET CEE Specifies the high-level qualifier for the LE runtime library. CEE should be set to the prefix of the IBM Language Environment data sets. Language Environment is required to run `crmlkon`.
- SET CBC Specifies the high-level qualifier for the C/C++ runtime library. CBC should be set to the prefix of the IBM C++ data sets.

Note: Uncomment the `SET CBC` line and tailor the `STEPLIB` concatenation if these libraries are not in your system link library concatenation.

Listing 6-1 Sample `crmlkon.jcl` for `crmlkon` Command

```
//*****
**
//* THIS JOB IS USED FOR THE STAND-ALONE LINK COMMAND          *
//* TO ACTIVATE A REMOTE LINK. SEE USER GUIDE FOR MORE INFO  *
//*                                                           *
//* @(#) $Id: crmlkon.jcl,v 1.10 2001/05/07 23:41:27 crount Exp $ *
//* Copyright (c)2000 BEA Systems, Inc., all rights reserved. *
//*****
**
//* YOU MUST SET THE ENVIRONMENT VARIABLES NEEDED BY CRMLKON *
//*****
**
//*****
**
//* LINKCMD INDICATES THE DISTRIBUTED SNACRM ADDRESS AND LINKNAME*
//* OBJLIB IS THE LOAD LIBRARY CONTAINING THE EAM PROGRAM OBJECTS*
```

```

/** RUNOPTS SETS ANY DESIRED LE RUNTIME OPTIONS *
/** DATA IS THE DATASET THAT CONTAINS THE ENVIRONMENT VARIABLES *
/** ENVFILE NAMES THE MEMBER THAT CONTAINS THE ENVIRONMENT VARS *
/** SIZE SETS THE REGION SIZE FOR THE SNACRM PROCESS *
/** TAILOR YOUR JCL FOR THE BELOW IF THESE LIBRARIES ARE NOT *
/** IN YOUR SYSTEM LINK LOAD LIBRARY CONCATENATION *
/** CEE IS THE HLQ FOR THE LE RUNTIME LIBRARY *
/** CBC IS THE HLQ FOR THE C/C++ RUNTIME LIBRARY *
/*******
**
// SET LINKCMD='-n<host name>:<port> <linkname>'
// SET OBJLIB=
// SET RUNOPTS=
// SET DATA=
// SET ENVFILE=ENV
// SET SIZE=1M
// SET ENV='ENVAR("_CEE_ENVFILE=DD:ENV")'
/** SET CEE=CEE,CBC=CBC
//CRMLKON EXEC PGM=CRMLKON,REGION=&SIZE,
// PARM='POSIX(ON) &ENV &RUNOPTS/&LINKCMD'
//STEPLIB DD DSN=&OBJLIB,DISP=SHR
/** DD DSN=&CEE..SCEERUN,DISP=SHR
/** DD DSN=&CBC..SCLBDLL,DISP=SHR
//ENV DD DSN=&DATA(&ENVFILE),DISP=SHR
//MSGFILE DD SYSOUT=*
//SYSPRINT DD SYSOUT=*

//

```

crmlkoff Command

The `crmlkoff` command stops one or more named CRM links.

`crmlkoff` stops all of the CRM links named on the command line. This is useful if one or more individual links need to be stopped after the CRM server booted. It can be used from any machine located on the same TCP/IP network as the machine running the CRM server. It can be used in a script and returns 0 if the command could be sent to the target CRM. It returns 1 if the command could not be sent to the target CRM.

Example of the `crmlkoff` command line:

```
crmlkoff -n<hostname:port> [-v -i] <linkname> ...
```

Command Line Options

The following table provides descriptions of the valid options for the `crmlkoff` command:

Keyword	Default	Required/Optional	Description
<code>-n<hostname:port></code>	None	Required	Names the machine and port running the CRM server.
<code>-v</code>	Off	Optional	Specifies verbose. Normally the command will not produce any messages, facilitating use in a script.
<code>-i</code>	Off	Optional	Ignores errors. When specifying multiple links, any error encountered when issuing CRM commands causes <code>crmlkon</code> to stop processing links and return. Errors can be ignored for individual links and processing continues with the next named link
<code><linkname></code>	None	Required	Names the link to be stopped. This is the <code>JC_SNALINKS</code> entry in the <code>jcrmgw.cfg</code> that defines this link. Multiple link names can be specified.

Example

To stop links `link1` and `cicstest` owned by the CRM running on `mach` at port 5000:

```
crmlkoff -n mach:5000 link1 cicstest
```

Diagnostics

`crmlkoff` only checks the syntax of the command. Use the CRM Monitor to determine if the link actually became active. Refer to [“Setting CRM Monitor Options”](#) for more information. If the command could not be successfully sent to the CRM, `crmlkoff` prints an error message if in verbose mode and exits with error code 1. Upon successful completion, `crmlkoff` exits with exit code 0.

Sample crmlkoff Command JCL for OS/390 MVS Platform

This section provides an explanation of the SET commands and an example of JCL that can be used when you run the crmlkoff command. The sample SET commands may not reflect the configuration of your system. You must customize the SET commands for your environment. Refer to your System Administrator for more information about your particular setup.

- SET LINKCMD Sets the crmlkoff command line parameters. Refer to [“Sample crmlkon Command JCL for OS/390 MVS Platform”](#) for more information about the command line parameters.
- SET OBJLIB Indicates the name of the PDSE library where the crmlkoff executable is installed.
- SET DATA Indicates the dataset containing the ENVFILE.
- SET ENVFILE Indicates the name of the PDS member that contains the environment variables for the crmlkoff. A sample member, ENV, is delivered with your product.
- SET SIZE Defines the region size for the running crmlkoff task.
- SET ENV Indicates the ENVFILE DD name.
- SET CEE Specifies the high-level qualifier for the Language Environment (LE) runtime library. CEE should be set to the prefix of the IBM LE data sets. Language Environment is required to run crmlkoff.
- SET CBC Specifies the high-level qualifier for the C/C++ runtime library. CBC should be set to the prefix of the IBM C++ data sets.

Note: Uncomment the SET CBC line and tailor the STEPLIB concatenation if these libraries are not in your system link library concatenation.

Listing 6-2 Sample crmlkoff.jcl for crmlkoff Command

```

/*****
**
/** THIS JOB IS USED FOR THE STAND-ALONE LINK COMMAND          *
/** TO DEACTIVATE A REMOTE LINK. SEE USER GUIDE FOR MORE INFO *
/**                                                              *
/** @(#) $Id: crmlkoff.jcl,v 1.10 2001/05/07 23:41:27 crount Exp $ *
/** Copyright (c) 2000 BEA Systems, Inc., all rights reserved.  *
/*****
**

```

```
//* YOU MUST SET THE ENVIRONMENT VARIABLES NEEDED BY CRMLKOFF *
//*****
**
//*****
**
//* LINKCMD INDICATES THE DISTRIBUTED SNACRM ADDRESS AND LINKNAME*
//* OBJLIB IS THE LOAD LIBRARY CONTAINING THE EAM PROGRAM OBJECTS*
//* RUNOPTS SETS ANY DESIRED LE RUNTIME OPTIONS *
//* DATA IS THE DATASET THAT CONTAINS THE ENVIRONMENT VARIABLES *
//* ENVFILE NAMES THE MEMBER THAT CONTAINS THE ENVIRONMENT VARS *
//* SIZE SETS THE REGION SIZE FOR THE SNACRM PROCESS *
//* TAILOR YOUR JCL FOR THE BELOW IF THESE LIBRARIES ARE NOT *
//* IN YOUR SYSTEM LINK LOAD LIBRARY CONCATENATION *
//* CEE IS THE HLQ FOR THE LE RUNTIME LIBRARY *
//* CBC IS THE HLQ FOR THE C/C++ RUNTIME LIBRARY *
//*****
**
// SET LINKCMD='-n<host name>:<port> <linkname>'
// SET OBJLIB=
// SET RUNOPTS=
// SET DATA=
// SET ENVFILE=ENV
// SET SIZE=1M
// SET ENV='ENVAR("_CEE_ENVFILE=DD:ENV")'
//* SET CEE=CEE,CBC=CBC
//CRMLKOFF EXEC PGM=CRMLKOFF,REGION=&SIZE,
// PARM='POSIX(ON) &ENV &RUNOPTS/&LINKCMD'
//STEPLIB DD DSN=&OBJLIB,DISP=SHR
//* DD DSN=&CEE..SCEERUN,DISP=SHR
//* DD DSN=&CBC..SCLBDLL,DISP=SHR
//ENV DD DSN=&DATA(&ENVFILE),DISP=SHR
//MSGFILE DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//
```

Using the CRM Monitor

The CRM Monitor is a Java application that allows you to connect to and monitor the CRM server through a graphical user interface (GUI). You can use the CRM Monitor to monitor link status and activity and start or stop diagnostic tracing of the CRM

server. The CRM Monitor makes a network connection to the remote CRM server through a TCP/IP network connection. The CRM Monitor application file is the `crmon.jar` file.

Launching the CRM Monitor from the Windows NT Desktop

If the CRM Monitor is installed on Windows NT, a short cut is created in the Start menu of the Desktop when you install JAM.

To launch the CRM Monitor from the Desktop:

- Choose Programs|BEA Weblogic E-Business Platform|Java Adapter for Mainframe|CRM Monitor from the Start menu.

Launching the CRM Monitor from the Command Line

The CRM Monitor can also be started from the command line on UNIX and Windows platforms.

To launch the CRM Monitor from the command line:

- Enter the following command for the Java 1.2.2 interpreter:

```
java -jar crmon.jar
```

Setting CRM Monitor Options

[Figure 6-1](#) shows an example of the CRM Monitor display.

Figure 6-1 CRM Monitor Display

The screenshot shows the CRM Monitor application window. The title bar reads "CRM Monitor". The main content area includes:

- CRM Address:** A text field containing "://dalvs5:8002".
- Selected CRM:** A text field displaying "Address //dalvs5:8002 - Group SC3GWY - Process 33554506 - flags: Configured".
- Trace Status:** A text field showing "Current CRM trace level is 0, App Stack trace is OFF".
- Trace Options:** A set of buttons for "Off", "Minimum", "Medium", "Maximum", "APPC Off", and "APPC On".
- Link Status:** A text area showing "Remote Links:" followed by a list of links and their statistics: "00 WEBL<->CICS SMSNA100 Ins Acq Max=10 Active= 5 InUse= 4 A06CICS 000 001 002 004".
- Link Statistics:** A text area showing "Statistics for Remote Link A06CICS" with input/output conversation and message counts.
- Messages:** A text field at the bottom showing "Signon to //dalvs5:8002 successful".

Callouts on the right side of the window provide detailed descriptions for each section:

- CRM Address:** Enter the address of the CRM to be monitored.
- Selected CRM:** Displays name of the CRM at the address.
- Trace Status:** Displays the currently selected trace option(s).
- Trace Options:** Select the trace option(s). One of four CRM options and/or one of two APPC options.
- Link Status:** Displays the current status of all remote links for the selected CRM. (Text may be scrolled.)
- Link Statistics:** Displays the current statistics for all remote links to the selected CRM. (Text may be scrolled.)
- Messages:** Displays results of automatic connection attempts and/or entered commands.

After you launch the CRM Monitor, use the following instructions to set CRM Monitor options:

1. To monitor a CRM, type the CRM address in the **Enter CRM Address** text box using the following format:

```
//host:port
```

where `host` explicitly specifies the CRM host machine and `port` specifies the port number of the CRM on the CRM host machine.

This value may only be entered once per started instance of a CRM Monitor.

2. Select one of the following trace options for the CRM:
 - **Stop CRM Trace** disables CRM tracing and closes the trace file, if it exists. This option is trace level 0 as described in the Trace Options section for the [“CRM Command”](#) command.
 - **Set Minimum CRM Trace** establishes tracing of only major events. This level is sufficient only to determine the sequence of application conversations. This option is trace level 1 as described in the Trace Options section for the [“CRM Command”](#) command.
 - **Set Medium CRM Trace** establishes minimum tracing plus tracing of all I/O buffers. This option is trace level 2 as described in the Trace Options section for the [“CRM Command”](#) command.
 - **Set Maximum CRM Trace** establishes medium tracing plus tracing of all APPC verbs. This option is trace level 3 as described in the Trace Options section for the [“CRM Command”](#) command.

Note: The CRM Monitor does not show trace data. This data is captured in a file under the `APPDIR` directory of the CRM server (`APPDIR` is the variable name associated with the CRM directory). Please contact BEA Customer Support for help in locating the trace file(s) and interpreting them.

3. Select one of the following APPC stack trace options:
 - **Start APPC Stack Trace** establishes tracing of the APPC stack. This option generally shows the parameters and results of all API calls. Depending on the stack being used, other options such as vendor-specified environment variables also may have to be activated. This option may be selected along with any of the previous Trace Options. This option corresponds to the `-s` option as described in the [“CRM Command”](#) section.

- **Stop APPC Stack Trace** disables APPC stack trace, if established.

Note: Trace options for the CRM and APPC Stack API trace options may also be set from the CRM command. Refer to the “[CRM Command](#)” section for more information about trace options.

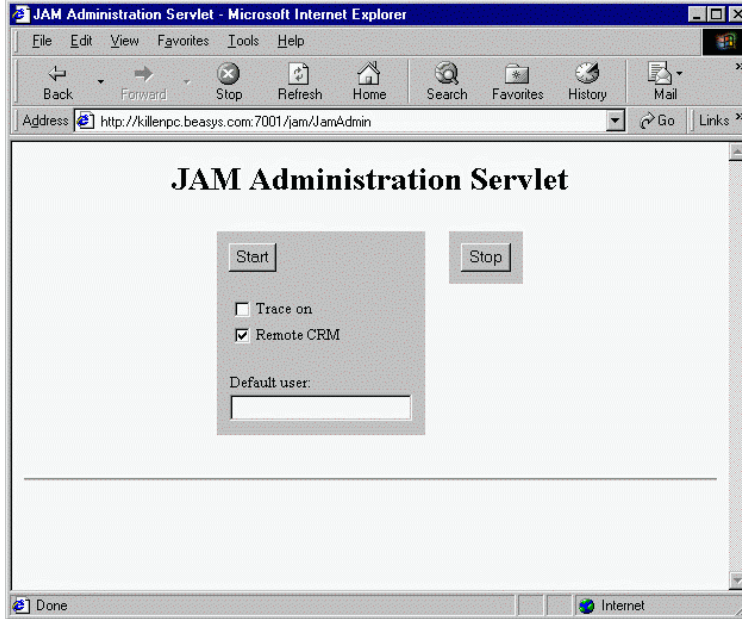
4. After you enter a CRM address and select trace options, the following fields display information about the CRM you are monitoring:

Display Field	Description
Selected CRM	Displays the name of the CRM at the address entered in the address field.
Trace Status	Displays the currently selected trace options.
Link Status	Displays the current status of all remote links for the selected CRM. (Text may be scrolled.)
Link Statistics	Displays the current statistics for all remote links for the selected CRM. (Text may be scrolled.)
Message Line	Displays messages showing either the results of automatic connection attempts or commands issued to change the trace options.

Starting and Stopping the Gateway with the JAM Administration Servlet

Figure 6-2 shows an example of the JAM Administration Servlet window:

Figure 6-2 JAM Administration Servlet Window



The JAM Administration Servlet is packaged in the `jam.ear` file from the JAM distribution. The servlet is available on any system where the JamAdmin Web application is deployed from the `jam.ear` file.

Launching the JAM Administration Servlet

To display the JAM Administration Servlet in a browser window, enter the following URL:

```
http://<host>:<port>/jam/JamAdmin
```

In this example, `<host>` and `<port>` specify the WebLogic listen address.

Setting JAM Administration Servlet Options

You may set several boot options to be used when the JAM gateway is started from the JAM Administration Servlet window. These options correspond to options used with the `GWBOOT` command. For more detailed information about the `GWBOOT` command options, refer to the [“Adding the JAM Startup Class to the WebLogic Server Configuration File”](#) section.

The following options may be set in the JAM Administration Servlet window before you start the JCRMGW gateway:

Option	Corresponding GWBOOT Option	Description
Trace on	-t	Turns on level 3 CRM tracing if started as a local CRM (no -r option)
Remote CRM	-r	Specifies a remote CRM
Default user	-u <name>	Specifies a user ID

Note: When the JAM Administration Servlet is started, the options selected, `Trace on`, `Remote CRM`, and `Default user`, reflect the current configuration as established by either the WebLogic Server Console or this servlet.

Using JAM Administration Servlet Start and Stop Commands

The following command buttons on the JAM Administration Servlet window allow you to start or stop JAM:

Command Button	Description
Start	Initializes the gateway with the options specified. The message 'JAM has been successfully started' appears if the operation succeeds.
Stop	Stops the gateway if it is currently running. The message 'JAM has been successfully stopped' appears if the operation succeeds.

What Do I Do Next?

In the normal course of operations your JAM system may occasionally need to be shut down for planned outages of your remote environment. Refer to the next section, [“Shutting Down the System for Planned Outages,”](#) for more information. Otherwise, you can begin using JAM for workflow processing or generating Java source code. If you want to perform workflow processing between Java applications and the mainframe, refer to the *BEA WebLogic Java Adapter for Mainframe Workflow Processing Guide*. If you want to develop Java applications by generating Java source code from COBOL copybooks, refer to the *BEA WebLogic Java Adapter for Mainframe Programming Guide*.

7 Shutting Down the System for Planned Outages

Occasionally, your JAM system may need to be shut down for planned outages of your remote environment. You may shut the JAM gateway and the CRM down together or independently without stopping or re-starting WebLogic Server. In most cases, you would probably not want to shut down the CRM independently. When you shut the gateway down, all links associated with the gateway are shut down. You can also shut down specific links without shutting down the gateway.

Action List

Before you shut down your JAM system for planned outages, see the following action list and refer to the appropriate information sources.

Table 7-1 Action List for Shutting Down the System for Planned Outages

Your action...	Refer to...
1 Complete all prerequisite tasks.	“Prerequisites”
2 Shut down the system.	“Shutting Down the System”
3 Restart the system.	“Restarting the System”

Table 7-1 Action List for Shutting Down the System for Planned Outages

Your action...	Refer to...
4 Determine what you want to do next.	<i>“What Do I Do Next?” BEA WebLogic Java Adapter for Mainframe Workflow Processing Guide, or BEA WebLogic Java Adapter for Mainframe Programming Guide</i>

Prerequisites

Before you shut down your JAM system for planned outages, you should complete the following tasks:

Your action...	Refer to...
1 Verify that system requirements have been met.	<i>BEA WebLogic Java Adapter for Mainframe Introduction and Release Notes</i>
2 Determine which configuration your system requires and install the JAM product as appropriate for your configuration.	<i>BEA WebLogic Java Adapter for Mainframe Installation Guide</i>
3 Verify that your mainframe configuration has been properly set up for the CRM.	“Preparing Mainframe Configurations for CRM Requirements”
4 Create a JAM configuration file and verify that the gateway has been properly configured.	“Defining the JAM Gateway Configuration”
5 Learn how to start or stop the CRM with the CRM administration commands.	“Using the CRM Administration Commands”
6 Deploy your configuration and verify that the gateway is working properly.	“Deploying Your Configuration”

Your action...	Refer to...
7 Learn about the JAM administration utilities.	“Using JAM Administration Utilities”

Shutting Down the System

To shut down all, or parts of your JAM system without stopping or re-starting WebLogic Server, perform the following steps:

1. Shut down your CRM, gateway and associated links, or one or more links, using the following procedures as required:

Component	Shutdown Procedure
CRM	Use the <code>crmdown</code> command. Refer to the “crmdown Command” and “Stopping the CRM On All Platforms” sections for more information. Most likely you will not want to shut down the CRM without shutting down the gateway.
Gateway and associated links	Stop the gateway from the JAM Administration Servlet. Refer to the “Starting and Stopping the Gateway with the JAM Administration Servlet” section for more information.
Link	Use the <code>crmlkoff</code> command. Refer to the “crmlkoff Command” section for more information.

2. Using a procedure as appropriate for your system, bring your CICS or IMS regions down.

Restarting the System

To restart all, or parts of your JAM system, perform the following steps:

1. Using a procedure as appropriate for your system, bring up your CICS or IMS regions.
2. Restart the JAM components as required using the following procedures:

Component	Restart Procedure
CRM	Use the CRM command. Refer to “CRM Command” and “Starting and Stopping the CRM” for more information.
Gateway and associated links	Start the gateway from the JAM Administration Servlet. Refer to the “Starting and Stopping the Gateway with the JAM Administration Servlet” section for more information.
Link	Use the crmlkon command. You can specify one or more links to start. Refer to the “crmlkon Command” section for more information.

What Do I Do Next?

After you configure the JAM gateway and verify that your JAM system is working properly, you can begin using JAM for workflow processing or generating Java source code. If you want to perform workflow processing between Java applications and the mainframe, refer to the *BEA WebLogic Java Adapter for Mainframe Workflow Processing Guide*. If you want to develop Java applications by generating Java source code from COBOL copybooks, refer to the *BEA WebLogic Java Adapter for Mainframe Programming Guide*.

Index

A

Advanced Peer-to-Peer Networking (APPN)
1-4

C

CICS DPL 2-14

CICS/ESA LU
configuration 1-15

commands

CRMDOWN 4-8

SET CEE 4-7

SET OBJLIB 4-7

communications resource manager
about 1-3

configuration

CICS/ESA LU 1-15

CRM

starting on OS/390 MVS 4-12

starting on UNIX or Windows NT 4-11

CRM Monitor

using 6-1

CRMDOWN commands

SET CBC 4-10

SET CEE 4-10

SET DATA2 4-10

SET ENVFILE 4-10

SET LNKCMD 4-10

SET OBJLIB 4-10

SET SIZE 4-10

crmlkoff command 6-7

CRMLKOFF commands

SET CBC 6-9

SET CEE 6-9

SET DATA2 6-9

SET ENV 6-9

SET ENVFILE 6-9

SET LNKCMD 6-9

SET OBJLIB 6-9

SET SIZE 6-9

crmlkon command 6-4

CRMLKON commands

SET CBC 6-6

SET CEE 6-6

SET DATA1 6-6

SET DATA2 6-6

SET ENV 4-10, 6-6

SET ENVFILE 6-6

SET LNKCMD 6-6

SET OBJLIB 6-6

SET SIZE 6-6

customer support x

customer support contact information x

D

documentation

conventions x

where to find it viii

E

e-docs Web Site viii

F

FUNCTION

- as optional keyword for remote services 2-16

J

JCRMGW

- sample configuration file 2-3
- setting address of associated SNACRM 2-6
- using GROUP parameter to correlate with a SNACRM 2-7

jcrmgw.cfg file 2-4

- verifying with configuration checker utility 2-19

jsnacrm utility 6-11

L

link

- start command 6-4
- stop command 6-7

links

- activating and de-activating 6-3

Local Area Network (LAN) 1-4

M

Mainframe Application Alias 2-5

Mainframe Applications

- mapping program names to EJB servers 2-13

- mapping program names to method names 2-15

P

printing product documentation ix

protocols 1-3

R

RDOM

- as keyword for remote services 2-16

related information ix

Remote LU

- setting alias for SNACRM partner application 2-12

S

servers

- SNACRM 4-3

SET SNACMD command 4-7

SNA

- maximum number of sessions per link 2-10
- setting minimum number of sessions as contention winners 2-11

SNACRM

- command 4-3
- identifying partner mainframe application regions 2-9
- setting address in associated JCRMGW 2-6
- specifying Local LU for 2-8
- using GROUP parameter to correlate with a JCRMGW 2-7

SNACRM commands

- SET CBC 4-7
- SET DATA2 4-7
- SET ENV 4-7
- SET ENVFILE 4-7
- SET OBJLIB 4-7
- SET SIZE 4-7
- SET SNACMD 4-7

support

- technical x

T

technical support x

TRANTIME

as optional keyword for remote services
2-18

V

VTAM

mode table 2-11

W

winner sessions 1-16